

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-339355

(43) 公開日 平成8年(1996)12月24日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 7 0		G 0 6 F 15/16	3 7 0 N
9/46	3 6 0		9/46	3 6 0 C

審査請求 未請求 請求項の数16 O L (全 38 頁)

(21) 出願番号 特願平8-94250

(22) 出願日 平成8年(1996)4月16日

(31) 優先権主張番号 4 2 4 1 3 7

(32) 優先日 1995年4月17日

(33) 優先権主張国 米国 (U S)

(71) 出願人 595119464

エイ・ティ・アンド・ティ・アイビーエム・コーポレーション

アメリカ合衆国, 33134 フロリダ, コーラル ゲーブルズ, ポンス ド レオン
ブウルヴァード 2333

(72) 発明者 フィリップ スティーヴン ウィンターボトム

アメリカ合衆国, 07933 ニュージャージー, ジレット, マウンテン アヴェニュー
591 アパートメント ナンバー 8

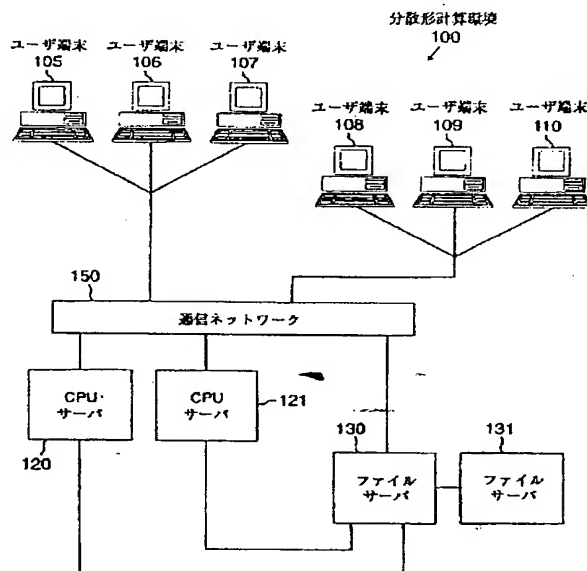
(74) 代理人 弁理士 三俣 弘文

(54) 【発明の名称】 分散形システムでの処理タスク実行呼び出し方法及び装置

(57) 【要約】

【課題】 分散形システムでの処理タスク実行呼び出し方法及び装置の改善。

【解決手段】 分散形計算システム100内の第1のプロセッサ105が行う、遠隔のプロセッサ120上での処理タスク(タスク)の実行を呼び出すための方法において、タスクが第1のプロセッサに連関する第1のネーム空間において動作し第1のネーム空間が複数のネーム空間修正コマンドからなり、この方法が、第1のプロセッサと遠隔のプロセッサとの間に通信リンク150を設立するステップと、第1のネーム空間を形成する、少なくとも1個の動的ネーム空間修正コマンドからなる複数のネーム空間修正コマンド、を特定するステップと、これら複数のネーム空間修正コマンドに基づいて修正されたネーム空間上で遠隔のプロセッサによってタスクを実行するために、これら複数のネーム空間修正コマンドを遠隔のプロセッサへ送信するステップとからなる。



【特許請求の範囲】

【請求項1】 分散形計算システム内の第1のプロセッサによって行われる、遠隔のプロセッサ上での処理タスクの実行を呼び出すための方法であって、該処理タスクが、該第1のプロセッサに関連する第1のネーム空間において動作し、該第1のネーム空間が、複数のネーム空間修正コマンドによって形成される、ようにした処理タスク実行呼び出し方法において、

該方法が、

前記第1のプロセッサと前記遠隔のプロセッサとの間に通信リンクを設立するステップと、

前記第1のネーム空間を形成する前記複数のネーム空間修正コマンドを特定するステップであって、特定された該複数のネーム空間修正コマンドが少なくとも1個の動的ネーム空間修正コマンドからなるようにした、前記複数のネーム空間修正コマンドを特定するステップと、前記複数のネーム空間修正コマンドに基づいて修正されたネーム空間上で前記遠隔のプロセッサによって前記処理タスクを実行するために、前記複数のネーム空間修正コマンドを前記遠隔のプロセッサへ送信するステップと、からなることを特徴とする、処理タスク実行呼び出し方法。

【請求項2】 前記方法において、

もし、前記複数のネーム空間修正コマンドのうちの1個のネーム空間修正コマンドに、リソースのファイルシステムを前記第1のネーム空間内の指示された部分へマウントする動作が含まれる場合に、該1個のネーム空間修正コマンドに、該マウントされたリソースにアクセスするために前記遠隔のプロセッサが利用する通信リンクを特定する動作が含まれるようにしたことを特徴とする請求項1の方法。

【請求項3】 前記方法において、

前記特定された通信リンクが、もし前記マウントされたファイルシステムが前記第1のプロセッサと同じネットワークノード上のリソースによって供給される場合に前記リソースと通信するために前記第1のプロセッサが利用するプロセス間通信パイプである、ようにしたことを特徴とする請求項2の方法。

【請求項4】 前記方法において、

前記特定された通信リンクが、もし前記マウントされたファイルシステムが前記第1のプロセッサと異なるネットワークノード上のリソースによって供給される場合に前記遠隔のプロセッサが前記リソースへのネットワーク接続を設立すべきことを表示する、ようにしたことを特徴とする請求項2の方法。

【請求項5】 前記方法において、

前記複数のネーム空間修正コマンドの各々が、マウントテーブルのエントリに格納され、該マウントテーブル内の該エントリの各々が、該エントリが形成された順序を表示する値を有し、

前記第1のネーム空間を形成する前記複数のネーム空間修正コマンドを特定する前記ステップが、前記マウントテーブルエントリから前記ネーム空間修正コマンドを、それらのコマンドが実行された順に取り出す、ようにしたことを特徴とする請求項1の方法。

【請求項6】 前記方法が更に、

前記第1のプロセッサと同じネットワークノード上に常駐するファイルについて、前記遠隔のプロセッサから前記第1のプロセッサによって受信されたファイル動作要求を前記遠隔のプロセッサに協力して処理するステップからなるようにしたことを特徴とする請求項1の方法。

【請求項7】 分散形計算システム内の遠隔のプロセッサによって行われる、第1のプロセッサから受信され呼び出された処理タスクを実行するための方法であって、該処理タスクが、該第1のプロセッサに関連する第1のネーム空間において動作し、該第1のネーム空間が、複数のネーム空間修正コマンドによって形成され、該遠隔のプロセッサが第2のネーム空間を有する、ようにした処理タスク実行方法において、

該方法が、

前記第1のプロセッサからの接続要求を処理するステップと、

前記呼び出された処理タスクを実行するためのプロセスを前記遠隔のプロセッサ上に割り当てるステップと、前記第1のネーム空間を形成する前記複数のネーム空間修正コマンドを前記第1のプロセッサから受信するステップであって、受信された該複数のネーム空間修正コマンドが少なくとも1個の動的ネーム空間修正コマンドからなる、ようにした受信するステップと、

前記第1のプロセッサへの通信リンクを設立するステップと、

前記第2のネーム空間内の予め定められた場所に前記第1のプロセッサへの前記通信リンクをマウントするステップと、

前記第2のネーム空間を修正するために、受信された前記複数のネーム空間修正コマンドを実行するステップと、

前記呼び出された処理タスクを、前記修正された第2のネーム空間上で実行するステップと、からなることを特徴とする、処理タスク実行方法。

【請求項8】 前記方法において、

もし、前記複数のネーム空間修正コマンドのうちの1個のネーム空間修正コマンドに、リソースのファイルシステムを前記ネーム空間内の指示された部分へマウントする動作が含まれる場合に、該1個のネーム空間修正コマンドに、該マウントされた特定のリソースにアクセスするために前記遠隔のプロセッサが利用する通信リンクを特定する動作が含まれるようにしたことを特徴とする請求項7の方法。

【請求項9】 前記方法において、

前記特定された通信リンクが、もし前記マウントされたファイルシステムが前記第1のプロセッサと同じネットワークノード上のリソースによって供給される場合に前記リソースと通信するために前記第1のプロセッサが利用するプロセス間通信パイプである、ようにしたことを特徴とする請求項8の方法。

【請求項10】 前記方法において、前記特定された通信リンクが、もし前記マウントされたファイルシステムが前記第1のプロセッサと異なるネットワークノード上のリソースによって供給される場合に、前記遠隔のプロセッサが前記マウントされたリソースへのネットワーク接続を設立すべきことを表示する、ようにしたことを特徴とする請求項8の方法。

【請求項11】 前記方法が更に、前記第1のプロセッサと同じネットワークノード上に常駐するファイルについて、ファイル動作要求を前記第1のプロセッサへ送信するステップからなるようにしたことを特徴とする請求項7の方法。

【請求項12】 複数のリソースからなる分散形計算システムのノード内にネーム空間情報を格納するための装置であって、該リソースの各々が階層形ファイルシステムとして表現され、該ネーム空間が、該ネーム空間に1個以上のネーム空間修正コマンドによって加えられた少なくとも1個の該リソース階層形ファイルシステムからなり、該ネーム空間がプロセス毎に修正可能であるようにした、ネーム空間情報格納装置において、

該装置が、前記ネーム空間修正コマンドによって前記ネーム空間内の指示された部分に加えられた前記リソース階層形ファイルシステムの各々についての表示を格納するためのマウントテーブルと、

前記ネーム空間に加えられた前記リソース階層形ファイルシステムの各々に連関するサーバ樹木バステ이블であって、該サーバ樹木バステ이블が、前記連関するリソース階層形ファイルシステム内の前記ファイルの各々についてのパス情報を供給し、該パス情報が、前記リソース階層形ファイルシステムの階層の再形成を可能にし、前記ファイルの各々についてのパス情報が、前記連関するファイルのネームと、前記連関するファイルの前記階層内の親のファイルの表示とからなるようにした、サーバ樹木バステ이블と、からなることを特徴とする、ネーム空間情報格納装置。

【請求項13】 前記装置において、前記マウントテーブルが、前記ネーム空間修正コマンドの各々についてのエントリからなり、前記マウントテーブル内の各エントリが、該エントリが形成された順序を表示する値からなる、ようにしたことを特徴とする請求項12の装置。

【請求項14】 前記装置において、前記マウントテーブルが、前記ネーム空間修正コマンド

の各々についてのエントリからなり、前記マウントテーブル内の各エントリが、前記ネーム空間内の第1のファイルの解釈が第2のファイルに翻訳されるべきことを表示する、ようにしたことを特徴とする請求項12の装置。

【請求項15】 前記装置において、前記サーバ樹木バステ이블が、複数のバスデータ構造にアクセスするための少なくとも1個のポインタを有し、該複数のバスデータ構造の各々が、前記連関するリソース階層形ファイルシステム内の1個のファイルに関連し且つ前記ファイルネームと前記親情報を格納する、ようにしたことを特徴とする請求項12の装置。

【請求項16】 分散形計算システムにおける与えられたチャンネルに連関するバスネームを特定するための方法であって、該分散形計算システムが、複数の分散形リソースを有し、該複数の分散形リソースの各々が、1個以上のファイルからなる階層形ファイルシステムとして表され、1個以上の該階層形ファイルシステムの集合がネーム空間を形成し、該階層形ファイルシステムの各々が、ルートディレクトリを有し、該チャンネルの各々が、該ネーム空間内の与えられたファイルを表し、該ファイルの各々が、英数字ラベルからなるファイルネームを表し、該バスネームが、該階層形ファイルシステムを通る該ルートディレクトリから該ファイルへのパスを指定することによって該ファイルを特定するようにした、分散形計算システムにおけるバスネームを特定するための方法において、

該方法が、前記チャンネルの各々に1個のバスデータ構造を連関させるステップであって、該バスデータ構造が、該連関するファイルのファイルネームと前記ネーム空間階層内の親のファイルに連関する前記バスデータ構造を指すポインタとを格納するようにした、連関させるステップと、バスネームを特定されるべきチャンネルの表示を受信するステップと、

前記表示されたチャンネルに連関するバスデータ構造を特定するステップと、

前記特定されたバスデータ構造から、前記ファイルネームと前記親のファイルに連関する前記バスデータ構造を指すポインタとを取り出すステップと、

前記親のファイルに連関する前記バスデータ構造を指す前記取り出されたポインタに従うとともに、前記階層形ファイルシステムの前記ルートディレクトリに到達するまで前記の、本ステップの前のステップを反復するステップであって、前記バスデータ構造から取り出された前記ファイルネームの各々が、前記表示されたチャンネルに連関する前記生成されたバスネームの先頭部に付けられるようにした、ポインタに従うとともに反復するステップと、からなることを特徴とする、バスネームを特定するための方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、分散形計算システムに関し、詳しくは、分散形計算システムにおけるネーム（名前）空間情報の格納及び取り出しに関する。

【0002】

【従来の技術】計算システムにおけるリソースと計算タスクとは複数のネットワークノード間にますます拡散しつつありこれによって分散形計算システムが形成されるようになって来ている。分散形計算システムがますます

人気を得ている大きな理由は、同システムによってコスト効率及び可用性が増大するためである。集中形リソースが分散形システム内の複数のユーザによって共用される場合、リソースのコストが、より大きなユーザ基盤にわたって分散される。

【0003】加えて、共用リソースの集中によって、これらリソースの管理及び保守がより効率的になり、又集中バックアップ機構が利用可能になることから潜在的に信頼性が増大する。更に、大抵の分散形計算環境によって得られる冗長性から、故障が検出された場合に代替のデバイス上でプロセス処理を続行できるため、故障に対する回復能力が改善される。

【0004】分散形計算システム（又は簡単に、分散形システム）は処理時間においても同様の改善が得られる。例えば或る特定の処理タスクを、分散形システムにおいてはいくつもの小さなタスクに分割して複数の分散形プロセッサによって同時に実行することが可能となる。加えて、個々の処理タスクの処理時間も、タスクをより強力な集中プロセッサへ移出（エクスポート）することによって、しばしば改善される。

【0005】しかし、分散形計算システムを成功させるには、ユーザが種々の且つ分散したリソースに一律すなわち同一の仕方でアクセスできることが必要である。したがって、与えられたリソースの物理的な場所及びそのプロトコルの特異な詳細は、ユーザには透過的（すなわち、ユーザが知らなくてよい状態）でなければならない。

【0006】AT&Tベル研究所によって開発された「プラン9」形分散処理システムのような、いくつもの分散計算環境において、システム内の各リソースは、そのリソースによって供給されるファイル類似の対象物（オブジェクト）の樹木からなる階層形ファイルシステムとして実現される。したがって、分散形システム内の異なる個々のリソースは、もしそれらアクセスされるリソースが従来の意味でのファイルではない場合でも、同一のファイル志向方式でアクセスすることが可能である。

【0007】一般に、このような分散形計算環境におけるオペレーティングシステムにより、各「ファイル」がそれに連関する英数字のネーム（名前）を持つことが可

能となる。この英数字ネームはネーム空間内のファイルにアクセスするために利用される。加えて、ファイル間の関係を指定し、これによってファイルをディレクトリとファイルとの階層形樹木に構成することが可能となる。ファイルは一般に、与えられたファイルへの階層形構造を通してのパスネーム（経路名）によってネーム空間において特定される。

【0008】分散形計算環境におけるネーム空間の全般的な説明については、文献（Roger M. Needham, "Names," (Distributed Systems 315-27) (Sape Mullender ed., 2ded. 1993))を参照されたい。

【0009】分散形システム内の各リソースをファイルセットとして表現することにより、望む透過性が得られ、各ユーザ又はプロセスは、利用可能なリソースセットについて自分に独自の構成図を組み立てることが可能となる。したがって、各ユーザ又はプロセスは、1個以上の選択されたリソースによって得られるファイル樹木からなる個々にそのユーザまたはプロセスに合わせた（カスタム化した）ネーム空間を持つことができる。

【0010】その後、ユーザ又はプロセスがカスタム化された階層形ファイル樹木を通してパスネームによって特定されるファイル上でのオペレーション（作業）を実現する場合に、パスネームが、作業対象とすべき特定の、ファイル類似の対象物に変換される。

【0011】ローカルオペレーティングシステムの核は、ローカルユーザ又はプロセスのカスタム化されたネーム空間内のネームと物理的对象物との間の関係を追跡できるが、一方、ユーザ又はプロセスが処理タスクを、分散システム内の別の、多分より強力な遠隔のプロセッサにエクスポートする場合に、ネーム空間変換問題が生じる。

【0012】もし遠隔のプロセッサが、エクスポートする側のユーザ又はプロセスによって組み立てられたのと正に同一のネーム空間においてその特定の処理タスクを行わない場合には、パスネームを、オペレーション対象とすべき適切なファイル類似の対象物に正しく変換されないことになる。

【0013】

【発明が解決しようとする課題】上記の説明から明らかのように、第1のプロセッサが遠隔のプロセッサによる処理タスクの実行を呼び出すことができ、且つこの遠隔のプロセッサが、この第1のプロセッサから受信されたネーム空間表現に基づいて修正変更（又は簡単に、修正）されたネーム空間上でこの処理タスクを実行するような、分散形計算環境が求められている。

【0014】なお又、分散形計算システムのノード内にネーム空間情報を格納するための装置であって、格納されたネーム空間情報が、ネーム空間の指示された部分に加えられたリソースによって得られる階層形ファイルシステムの表示と、接続された各リソースのファイルシ

テム階層の再形成を可能にするバス情報とからなるような、ネーム空間情報を格納するための装置が求められている。

【0015】更に又、分散形計算システム内で、ファイルを表示する与えられたチャンネルに関連するバスネームを特定するための方法が求められている。

【0016】

【課題を解決するための手段】概して、本発明の一態様によれば、複数のコンピュータ及びネットワークにかけて分散された計算環境であって或る場所に所在するユーザが別の場所に所在するリソースにアクセスできるような分散された計算環境が提供される。この分散された計算環境により、第1のプロセッサが、中央処理装置のような遠隔のプロセッサによる処理タスクの実行を呼び出すことが可能となる。

【0017】本発明の一態様によれば、この第1のプロセッサがその現ネーム空間の表現を遠隔のプロセッサへ送信する。遠隔のプロセッサが、第1のプロセッサから受信されたネーム空間表現に基づいて変更されたネーム空間上で処理タスクを実行する。

【0018】本発明の別の態様によれば、分散形計算システム内の第1のプロセッサによって行われる、遠隔のプロセッサ上での処理タスクの実行を呼び出すための方法であって、第1のプロセッサが、複数のネーム空間変更コマンドによって形成される、連関する第1のネーム空間を有するような、処理タスク実行呼び出し方法が提供される。本処理タスク実行呼び出し方法は、第1のプロセッサと遠隔のプロセッサとの間に通信リンクを設立するステップからなる。

【0019】本方法は更に、第1のネーム空間を形成する複数のネーム空間変更コマンドを特定するステップであって、特定された複数のネーム空間変更コマンドが少なくとも1個の動的ネーム空間変更コマンドからなるようにした特定するステップと、複数のネーム空間変更コマンドに基づいて変更されたネーム空間上で遠隔のプロセッサによって処理タスクを実行するために、複数のネーム空間変更コマンドを遠隔のプロセッサへ送信するステップと、からなる。

【0020】本発明のなお別の態様によれば、分散形計算システム内の遠隔のプロセッサによって行われる、第1のプロセッサから受信され呼び出された処理タスクを実行するための方法であって、処理タスクが、第1のプロセッサに関連する第1のネーム空間において動作し、第1のネーム空間が、複数のネーム空間変更コマンドによって形成され、遠隔のプロセッサが第2のネーム空間を有する、ようにした処理タスク実行方法が提供される。

【0021】本処理タスク実行方法は、第1のプロセッサからの接続要求を処理するステップと、呼び出された処理タスクを実行するためのプロセスを遠隔のプロセ

サ上に割り当てるステップと、第1のネーム空間を形成する複数のネーム空間変更コマンドを第1のプロセッサから受信するステップであって、受信された複数のネーム空間変更コマンドが少なくとも1個の動的ネーム空間変更コマンドからなる、ようにした受信するステップとからなる。

【0022】本処理タスク実行方法は更に、第1のプロセッサへの通信リンクを設立するステップと、第2のネーム空間内の予め定められた場所に第1のプロセッサへの通信リンクをマウントするステップと、第2のネーム空間を変更するために、受信された複数のネーム空間変更コマンドを実行するステップと、呼び出された処理タスクを、変更された第2のネーム空間上で実行するステップと、からなる。

【0023】本発明の更に別の態様によれば、複数のリソースからなる分散形計算システムのノード内にネーム空間情報を格納するための装置であって、リソースの各々が階層形ファイルシステムとして表現され、ネーム空間が、ネーム空間に1個以上のネーム空間変更コマンドによって加えられた少なくとも1個のリソース階層形ファイルシステムからなり、ネーム空間がプロセス毎に変更可能であるようにした、ネーム空間情報格納装置が提供される。

【0024】本ネーム空間情報格納装置は、ネーム空間変更コマンドによってネーム空間内の指示された部分に加えられたリソース階層形ファイルシステムの各々についての表示を格納するためのマウントテーブルと、ネーム空間に加えられたリソース階層形ファイルシステムの各々に連関するサーバ樹木バステーブルとからなる。サーバ樹木バステーブルは、連関するリソース階層形ファイルシステム内のファイルの各々についてのバス情報を供給し、バス情報が、リソース階層形ファイルシステムの階層の再形成を可能にする。

【0025】ファイルの各々について格納されたバス情報が、連関するファイルのネームと、ネーム空間階層内の連関するファイルの階層内の親のファイルの表示とからなるようにした、サーバ樹木バステーブルとからなる。

【0026】本発明のなお別の態様によれば、分散形計算システムにおける与えられたチャンネルに関連するバスネームを特定するための方法が提供される。同方法は、分散形計算システムが、複数の分散形リソースを有し、複数の分散形リソースの各々が、1個以上のファイルからなる階層形ファイルシステムとして表され、1個以上の階層形ファイルシステムの集合がネーム空間を形成し、階層形ファイルシステムの各々が、ルートディレクトリを有する。

【0027】又、チャンネルの各々が、ネーム空間内の与えられたファイルを表し、ファイルの各々が、英数字ラベルからなるファイルネーム（ファイル名）を有し、バ

スネームが、階層形ファイルシステムを通るルートディレクトリからファイルへのパスを指定することによってファイルを特定する。

【0028】同方法は、チャンネルの各々に1個のバスデータ構造を連関させるステップであって、バスデータ構造が、連関するファイルのファイルネームとネーム空間階層内の親のファイルに連関するバスデータ構造を指すポインタとを格納するようにした、連関させるステップと、バスネームを特定されるべきチャンネルの表示を受信するステップと、記表示されたチャンネルに連関するバスデータ構造を特定するステップとからなる。

【0029】同方法は又、特定されたバスデータ構造から、ファイルネームと親のファイルに連関するバスデータ構造を指すポインタとを「検索して取り出す」（又は簡単に、「取り出す」と記載）ステップと、親のファイルに連関するバスデータ構造を指す取り出されたポインタに従うとともに、階層形ファイルシステムのルートディレクトリに到達するまでの、本ステップの前のステップを反復するステップであって、バスデータ構造から取り出されたファイルネームの各々が、表示されたチャンネルに連関する生成されたバスネームの前部に付けられるようにした、ポインタに従うとともに反復するステップとからなる。

【0030】本発明並びにその更なる機能及び利点について、より完全な理解を得るために、以下、図面を参照して詳細に説明する。

【0031】

【発明の実施の形態】本発明の機能を実現するのに適した分散形計算環境100の例示構成を示す。下で更に述べるように、分散形計算環境100は望ましくは、複数のコンピュータ及びネットワークにわたって分散し、1つの場所に所在するユーザの、別の場所に所在するリソースへのアクセスを、例えばユーザの端末と望むリソースとの間に通信リンクを設立することによって可能とするものである。

【0032】この分散形計算環境100により、処理タスクの遠隔のプロセッサへのエクスポートが可能となる。本発明の一態様によれば、ユーザまたはプロセスが、処理タスクを遠隔のプロセッサへエクスポートすべきであると表示した場合にエクスポートする側のユーザまたはプロセスに連関するネットワーク空間の現在の状態が点検評価される。

【0033】その後、ネットワーク空間の現在の状態、いい替えれば、ネーム空間を形成するためにユーザまたはプロセスによって実行されたネーム空間変更コマンドのシーケンスが、遠隔のプロセッサへ送信される。このようにして、遠隔のプロセッサがエクスポートする側のユーザまたはプロセスのネーム空間を正確に再形成することができ、これによってファイルネームを、運用すべき適切なファイル又はファイル類似の対象物に信頼性を

持たせて変換することができる。

【0034】本発明を以下8項目に分けて説明する。1番目に、まず「システムアーキテクチャ」の項で、図1及び図2に関連して分散形計算環境（すなわちネットワーク環境）全体及び分散形ネットワーク環境100の例示ノードについて述べる。

【0035】2番目に「ネーム空間におけるリソースの概念的表現」の項で、分散形計算システムにおけるユーザ又はプロセスに連関するネットワーク空間について述べる。この項においては、1個以上の遠隔のリソースの階層形ファイル樹木を特定のユーザ又はプロセスのカスタム化されたネーム空間を、望む場所へ連関させる仕方について説明する。

【0036】3番目に、「オペレーティングシステム核」の項で、本発明の機能を組み込んだ推奨オペレーティングシステムについて述べる。この項においては、オペレーティングシステム核によって分散形計算環境内の各リソースに与えられるインタフェースについて説明する。

【0037】4番目に、「分散形システムにおけるリソースへのアクセス」の項で、ネーム空間においてアクセスされる個々のファイル類似の対象物の表現についての推奨実現例について述べる。望ましくはアクセスされる各ファイルを表すような、チャンネルデータ構造の推奨実現例について図7に関連して説明する。加えて、ファイルアクセス情報を格納するために核によって利用されるデータ構造についても図6に関連して説明する。

【0038】5番目に、「非核ファイルシステムとの通信」の項で、核によって供給されないリソースと通信するために核によって設立されるデータ構造及び通信リンクについて述べる。この項においては、望む非核ファイルシステムへのパイプ又はネットワーク通信のような通信リンクの設立、非核ファイルシステムについてのアクセス及びバス情報を格納するための適切なデータ構造の形成、及びローカルユーザ又はプロセスのネーム空間への、非核ファイルシステムのファイル樹木ルートディレクトリのマウントの仕方、について説明する。

【0039】加えて、ローカルネーム空間の現在の状態を形成するネーム空間変更コマンドのシリーズを格納するためのマウントテーブル800の推奨実現例について、図8（B）及び図9に関連して説明する。

【0040】6番目に、「ネットワークの表現」の項で、ネットワーク形成及び維持についての推奨実現例について述べる。この項においては、記号的なマシンネームと、連関するネットワーク宛先アドレスとの間の翻訳について説明する。7番目に、「ネーム空間情報データ構造」の項で、本発明の機能を組み込んだいくつかの推奨バスデータ構造について述べる。

【0041】この項においては、特定のチャンネルに連関するバスネームをどのようにして定めることができる

か、及び接続された非核ファイルシステムの各々のファイル樹木内のアクセスされたファイルのネーム空間をどのようにして再形成するか、について説明する。

【0042】最後に、[ネーム空間プロセス]の項で、本発明の機能を組み込んだプロセスについて述べる。この項においては、処理タスクを、ローカルネーム空間の現在の状態の正確な表現と共に、遠隔のプロセッサへエクスポートするのに利用されるプロセス実行エクスポート1600について、図18に関連して説明する。

【0043】加えてローカルノードによってエクスポートされる処理タスクをインポートするために、図19に関連して述べるプロセス実行インポート1650が、選択された遠隔のプロセッサによって実行される。プロセス実行インポート1650がローカルノードのネーム空間を再形成し、これにより、ファイルネームが、運用すべき適切なファイル又はファイル類似の対象物に確実に正確に変換される。

【0044】図1に示す分散形計算環境100は望ましくは、AT&Tベル研究所(600 Mountain Avenue, Murray Hill, New Jersey 07974-0636)によって開発され、ネーム空間情報を格納しそして取り出すためのメカニズムを得るために変更を加えた「プラン9」形の分散形処理システムとして実施される。

【0045】「プラン9」形の分散形処理システムの詳細説明については、米国特許出願(出願番号08/237,967)(1994年5月3日出願)(現在放棄されている米国特許出願(出願番号07/702,651)(1991年5月17日出願)の継続出願)及び米国特許出願(出願番号08/247,626)(1994年5月20日出願)(現在放棄されている米国特許出願(出願番号07/999,755)(1992年12月31日出願)の継続出願)を参照されたい。これら2件の米国特許出願をここに本出願の参考文献とする。

【0046】又、「プラン9」形の分散形処理システムの追加説明が、文献(David Presotto et al., "Plan 9, A Distributed System," Proc. of the Spring 1991 EuroOpen Conference, pp. 43-50 (Tromsø, Norway 1991) 及び文献("Plan 9 from Bell Labs, Programmers Manual")にある。これら2件の文献をここに本出願の参考文献とする。これらの文献は、2個のディレクトリ((pub/plan9doc)及び(pub/plan9dist))内のインターネットアドレス(plan9.att.com)からファイルトランスファプロトコル(ftp)によって入手可能である。

【0047】[システムアーキテクチャ] 概して、分散形計算環境100は、端末105~110のような複数のユーザ端末と、1個以上の中央処理装置(CPU)サーバ120~121と、1個以上のファイルサーバ130~131とからなる。ユーザ端末105~110は、例えば図2(A)及び図(B)に関連して下で更に述べ

る、ミニコンピュータ、ワークステーション、又はパーソナルコンピュータのような汎用計算システム200として実施される。

【0048】下で更に述べるように、ユーザ端末105~110は望ましくは、分散形計算環境100の各ユーザに、ウィンドウシステム実行用のビットマップ画面及びマウスを有する専用コンピュータを提供する。このウィンドウシステムは、分散形計算環境100内のリソースへの多重化インタフェースを提供する。

【0049】CPUサーバ120~121は望ましくは、計算パワーを既知の仕方で大形マイクロプロセッサに集中させ、例えばシリコングラフィックス社から商業的に入手可能なパワーシリーズのマシンとして実施される。したがって、コンパイレーションのような集中計算タスクについて、ユーザは、図18に関連して下で更に述べる、次のコマンド処理を符号120又は121のようなCPUサーバに転送するCPUコマンドを実行することが可能となる。

【0050】CPUコマンドの実行に続いてユーザ端末105~110のアクティブウィンドウにコマンドインタープリタからのプロンプトが現れはするがコマンドインタープリタの実行動作(ラン)はCPUサーバ120又は121上で行われる。

【0051】下で述べるように、本発明により、遠隔のCPUサーバ120又は121によって行われる処理タスクを、ユーザ端末で実行中のユーザ又はプロセスのネーム空間と同一のネーム空間内で実行することが可能となる。ネーム空間については、下の[ネーム空間におけるリソースの概念的表現]の項で説明する。

【0052】ファイルサーバ130~131は、ユーザ端末105~110によって又CPUサーバ120~121によって既知の仕方で行われるファイルの格納保管のための保管場所を提供する。図3(A)及び図3(B)に関連して下で更に述べるように、ファイルサーバ130~131は、これらのファイルサーバ130~131によって与えられるファイルの樹木からなる階層形ファイルシステムを提供する。

【0053】ネットワークの組織と管理は、図1に示す分散形計算環境のような、分散形システムのどれにとっても重要な因子を構成する。このことは、CPUサーバ120~121及びファイルサーバ130~131のような、分散形計算環境100の重要構成要素が集中した場所に位置する場合に特にそうである。

【0054】一推奨実施例において、CPUサーバ120~121とファイルサーバ130~131とが、高帯域幅2点間光ファイバリンクのような高速ネットワーク接続によって相互接続される。しかしユーザ端末105~110は、イーサネット又はデータキット接続のような、上記の高速ネットワークよりも低速の通信ネットワーク150との接続によって分散形計算環境100内の

他のリソースに連結される。ネットワーク接続の形成及び管理についての適切な一例を、図10(A)及び図10(B)に関連して下で更に述べる。

【0055】ユーザ端末105~110として用いるのに適した汎用計算システム200の一実施例を、図2

(A)に更に詳細に示す。汎用計算システム200は例えば、ミニコンピュータ、ワークステーション、又はパーソナルコンピュータである。汎用計算システム200は望ましくは、表示モニタ204、処理装置206、ランダムアクセスメモリ(RAM)のような1個以上のメモリ装置207、バス224、並びにマウス208及びキーボード205のような1個以上のユーザインタフェースからなる。

【0056】上に述べたように、表示モニタ204は望ましくは、ウインドウシステムをサポートする。処理装置206及びメモリ装置207については図2(B)に関連して下で更に述べる。

【0057】汎用計算システム200は又、外部ディスクドライブ202及びハードディスクドライブ203からなる。外部ディスクドライブ202は、フロッピーディスク、テープ、又はコンパクトディスクのような1個以上の外部データ記憶装置を受け入れて読み出し及び書き込みを行うように作動する。一方、ハードディスクドライブ203は、高速アクセスデータ格納及び取り出し機能を既知の仕方で提供するように作動する。

【0058】加えて、汎用計算システム200は望ましくは、遠隔の装置とのシリアル又はパラレルデータ通信のための1個以上のデータ通信ポートを有する。例えば、汎用計算システム200は更に、フォアシステムズ社(Fore Systems, Inc. of Pittsburgh, Pennsylvania)から商業的に入手可能なATMホストアダプタカードのような、ネットワークインタフェース209に接続される。

【0059】図2(B)は、図2(A)の汎用計算システム200に関連して利用されるマイクロプロセッサ処理システムの概略ブロック図を示す。マイクロプロセッサ処理システムは、RAMのようなメモリ装置207にバス270を介して結合された処理装置206とネットワークインタフェース209とからなる。処理装置206は、単一のプロセッサ又は並行して作動するいくつかのプロセッサとして実現される。

【0060】メモリ装置207は、処理装置206が取り出し、解釈し、実行するように作動する対象となる1個以上の命令を記憶する。加えて、メモリ装置207は図4(A)、図4(B)、及び図5に関連して下で更に述べるオペレーティングシステムに連関するプログラムコード275を記憶する。

【0061】処理装置206は、制御装置250と、算術論理装置255と、例えばキャッシュ又は複数のレジスタのような、ローカルメモリ装置260とからなる。

制御装置250は、メモリ装置207から命令を取り出すように作動する。算術論理装置255は、命令を遂行するために必要な複数の演算を行うように作動する。ローカルメモリ装置260は、仮の結果及び制御情報を記憶するのに用いられる高速記憶装置を提供するように作動する。

【0062】[ネーム空間におけるリソースの概念的表現]前に述べたように、分散形計算環境100により、ユーザが複数のローカル及び遠隔のリソースにアクセスすることが可能になる。望ましくは、システム内の各リソースは、リソースによって供給されるファイル類似の対象物の樹木からなる階層形「ファイル」システムとして実現される。このようにして、ユーザは、アクセスされるリソースが従来の意味でのファイルではない場合でも、同一の、ファイル志向の仕方で各リソースにアクセスする。

【0063】以下の説明において、用語「ファイル」には従来の意味でのファイルと他のファイル類似の対象物を含む。なお、ユーザ又はプロセスのネーム空間内の各ファイルは、図6及び図7に関連して下で述べるチャネルデータ構造として表される。

【0064】すなわち、階層形ファイルシステムが、ファイルサーバ130~131によって与えられるような従来形ファイルシステム及び物理的デバイス及びプロセスのような他の非従来形リソースを表すために利用される。例えば、RS232ラインが、「データ」ファイルと「制御」ファイルとを含むディレクトリからなる階層形ファイルシステムとして表される。データが「データ」ファイルに書き込まれると、そのデータは連関するRS232ライン上で伝送される。

【0065】同様に、RS232ライン上で受信されたデータは、「データ」ファイルから読み出しを行うことによって取得される。「制御」ファイルには、RS232ラインに連関する制御パラメータが含まれ、これを修正することによって、ビットレートのような、物理的デバイスのパラメータが変更される。

【0066】オペレーティングシステム275により、ファイルシステム内の各ファイルがそれに連関する英数字ネーム又はラベルを持つことが可能になる。この英数字ネーム又はラベルを後にユーザが利用してネーム空間内でのそのファイルの所在場所が求められる。加えて、ユーザはファイル間の関係を指定して、これによりファイルをディレクトリ及びファイルの階層形樹木の形に組織化することができる。なお、ディレクトリは実際には、他のファイルの場所についての情報を含むファイルである。

【0067】概念的には、ディレクトリ及びファイルの階層形樹木より、ネームとエンティティ(主体)との間の関係を維持するためのメカニズムが得られ、これにより、ユーザがパスネームによって対象物の場所を求める

ことが可能になる。ファイルのパスネームは、階層構造を通してのファイルへのパスを指定することによってファイルを特定する英数字文字列として定義される。

【0068】一実施例において、パスネームは、ルートディレクトリを表す斜線（スラッシュ）文字“/”と、ルートディレクトリと望むファイルとの間の全てのディレクトリのネーム（斜線文字で互いに分けられている）とからなる。

【0069】ユーザ又はプロセスは、ユーザ又はプロセスについての連関するカスタム化されたネーム空間を形成するために1個以上の望むリソースを選択することにより分散形計算環境100内で利用可能なリソースセットについて個々に独自の構成図を組み立てる。例えば、図3（A）に示すような、ユーザ端末305上で作業中のユーザが、例えば個人的ファイルが常駐する或るファイルサーバと、特定のグループプロジェクトについてのソフトウェアが書き込まれる部門ファイルサーバとに接続されることを希望し、そのような独自の構成図を組み立てる場合が考えられる。

【0070】オペレーティングシステム275は一般に、ユーザ端末305のような端末に、端末がブートアップされる都度、最小デフォルトネーム空間320を供給する。デフォルトネーム空間320は例えば、入出力サービス、ユーザ端末305上でプロセッサを走らせるためのバイナリコード、及びプロセスサービスからなる。例示の「プラン9」形分散処理システム内のデフォルト又は「スタブ」ネーム空間についての更に詳細な説明は、例えば上に参考文献とした米国特許出願（出願番号08/237,967）（1994年5月3日出願）を参照されたい。

【0071】しかし、本明細書の説明においては、例示として、図3（A）に示すユーザ端末305のデフォルトネーム空間320は、2個のファイル、X及びYを含む1個のルートディレクトリ（斜線文字“/”で表す）だけからなるものとして述べる。

【0072】ファイル、X及びYは、図5及び図8（A）に関連して下で更に述べるように、他のリソースによって供給されるファイルシステムがマウントされる場所であるマウントポイントとして機能する。図3

（A）に示すように、ファイルサーバ310はネーム空間325を供給し、ファイルサーバ315はネーム空間330を供給する。

【0073】ユーザ端末305は、ネーム空間320内のファイルXを、ファイルサーバ310によって供給されるネーム空間325のルートディレクトリ“/”に等しくすることによって又ネーム空間320内のファイルYを、ファイルサーバ315によって供給されるネーム空間330のルートディレクトリ“/”に等しくすることによって、デフォルトネーム空間320を修正して、図3（B）に示す修正された修正ネーム空間320'を

形成することができる。

【0074】各ファイルサーバ310及び315をそれぞれユーザのローカルのネーム空間320'に結合させるには、ネットワーク340を通して適切な接続を設立する必要がある。ネットワーク接続を設立するため及び特定の遠隔のサービスのネーム空間をユーザ端末305に連関するネーム空間320内の望ましい場所に結合させるためのメカニズムを、図8（A）に関連して下に述べる。

【0075】なお、オペレーティングシステム275によって各ユーザに供給されるデフォルトネーム空間に加えて、個々のユーザは一般に、ネーム空間「プロフィール」に書き込みを行うことによってそのユーザのネーム空間をカスタム化することができる。ユーザによってカスタム化された「プロフィール」は、デフォルトネーム空間内のそのユーザの個人的ディレクトリ内に格納されるファイルで、ユーザが自動的に接続されるべきサービスのリソースのリストを包含する。

【0076】例えば、ユーザの「プロフィール」は、ユーザによって利用されるホームファイルサーバを表示し、ホームファイルサーバのルートがユーザのローカルのネーム空間“/”に接続されるべきことを表示する。加えて、ユーザの「プロフィール」は、デフォルトネーム空間に見出されるディレクトリ“/proc”にプロセスファイルシステムが接続されるべきことも表示する。

【0077】したがって、分散形計算環境100にログインすると、個々のユーザは、ユーザのカスタム化された「プロフィール」によって修正されたデフォルトネーム空間（以下包括的に、ユーザの予め定められたネーム空間、と称する）を自動的に与えられる。ユーザの予め定められたネーム空間は、デフォルトネーム空間とそのユーザのカスタム化されたネーム空間とに連関する一連の予め定められたネーム空間修正コマンドを実行することによって形成される。

【0078】加えて、操作中にユーザは、ユーザの予め定められたネーム空間を更に修正するように機能する動的ネーム空間修正コマンドを実行する。なお、動的ネーム空間修正コマンドをここに、ユーザ又はプロセスによって実行されるネーム空間修正コマンドであって、ユーザの予め定められたネーム空間に連関する予め定められたネーム空間修正コマンドのリストに含まれていないネーム空間修正コマンドを含むものと定義する。

【0079】すなわち、動的ネーム空間修正コマンドには一般に、分散形計算環境100にログイン後にユーザ又はプロセスによって実行されるネーム空間修正コマンドがどれも含まれる。

【0080】例えば、例示の「プラン9」形の分散形処理システムにおいては、1対のネーム空間修正コマンド、すなわち「bind」（バインド）（連結）コマンド及び「mount」（マウント）（搭載）コマンドが

10

20

30

40

50

与えられる。

【0081】「連結」コマンドは、チャンネルデータ構造として表現される現在のローカルのネーム空間内のファイル又はディレクトリ（以下、「from」ファイルと称する）を、別のチャンネルデータ構造として表される現在のネーム空間の別の部分（以下、「to」ファイルと称する）に等しくする。いい替えれば、「to」チャンネルは「from」チャンネルと連結された状態となる。

【0082】したがって、「連結」コマンドの実行後には、「from」バスネームによって指定されたファイルへの参照は、「to」バスネームによって指定されたファイルへの参照として解釈されることになる。

【0083】「マウント」コマンドは、チャンネルとして表される現在のローカルのネーム空間内に既に所在するネーム（以下、「from」ファイルと称する）を、別のチャンネルとして表される、遠隔のファイルサーバのような非核ファイルシステムのルート（以下、「to」ファイルと称する）に等しくする。いい替えれば「マウント」コマンドは、非核ファイルシステムのルートディレクトリを表すチャンネルを、現在のネーム空間の、指示された部分を表すチャンネル上にマウントする。

【0084】なお、ルートディレクトリにより、非核ファイルシステムによって与えられる階層形ファイル樹木全体へのアクセスが与えられる。このため、「マウント」コマンドの実行後には、「from」バスネームによって指定された、「自分の上に他からのマウント処理が行われた」（以下、「他からマウントを受けた」とも表現する）ファイル及び階層内でのそれより下方のファイルへの参照は、非核ファイルシステムのファイル樹木内のファイルへの参照として解釈されることになり、したがって、通信ストリームへ書き込まれた要求メッセージとなる。

【0085】図3（A）の例では、遠隔のファイルサーバ310のルートディレクトリ「/」がユーザ端末305のデフォルトネーム空間320内のファイルX上にマウントされている。

【0086】加えて、「連結」コマンド及び「マウント」コマンドは望ましくは、「from」ファイルと「to」ファイルとの間の関係についていくつかの異なる種類の関係を指定することが可能である。

【0087】まず、ユーザ又はプロセスは望ましくは、「to」ファイルによって参照されるもの全てによって、「from」バスネームによって参照されるもの全てが置換されるように指定することが可能である。加えて、ユーザ又はプロセスは、「from」バスネーム及び「to」バスネームが「from」ディレクトリファイルと「to」ディレクトリファイルとの両方にあるファイルからなる合同ディレクトリを形成するように指定することが可能である。

【0088】本質的に、この合同ディレクトリはいくつ

ものマウントされたサービスを有するディレクトリである。望ましくは、ユーザ又はプロセスは、この合同ディレクトリにおいて「from」ディレクトリを「to」ディレクトリの前に探すか又は後に探すかを指定することができる。

【0089】すなわち、ユーザ又はプロセスに関連するネーム空間の現在の状態は、ユーザによってカスタム化された「プロフィール」によって修正された、デフォルトネーム空間320に関連する予め定められたネーム空間修正コマンドのリストによって、又ユーザ又はプロセスによって実行された動的ネーム空間修正コマンドによって定められる。

【0090】なお、ユーザの予め定められたネーム空間を形成するデフォルトネーム空間及びユーザによってカスタム化された「プロフィール」は一般に、ユーザ又はプロセスのネーム空間内にファイルとして格納される。

【0091】本発明の一態様により、次に実行される「マウント」コマンド及び「連結」コマンドのような動的ネーム空間修正コマンドのリストは、図8（B）及び図9に関連して下で更に述べるような、各「マウント」コマンド及び「連結」コマンドのエントリを含むマウントテーブル800内に格納される。

【0092】望ましくはマウントテーブル800により、「マウント」コマンド及び「連結」コマンドのリストをこれらのコマンドの実行順にマウントテーブル800から取り出すことが可能となる。しかし、注記したいのは、マウントテーブル800から取り出された動的ネーム空間修正コマンドのリストが、バインド（連結）され又はマウント（搭載）されたその特定のチャンネルデータ構造によって表されるが、より有益なファイルバスネームでは表されないことである。

【0093】このため、下で述べる本発明の別の態様により、図12（A）～図12（C）に関連して述べるような、各チャンネルに対応するバスネームを定めることを可能にするバス情報を格納する付加的データ構造が設けられる。加えて、下で述べるように、格納されたバス情報により、接続された非核ファイルシステムの各々についての樹木内のアクセスされたファイルのネーム空間を再形成することが可能になる。

【0094】【オペレーティングシステム核】推奨する「プラン9」形の分散形処理システムのような分散形処理システムにおいて、オペレーティングシステム275が、図4（A）に示す核400として構成される。核400は一般に、分散形計算環境100内の核リソースへのインタフェースを提供する。

【0095】概念的には、核400は、プロセス管理ソフトウェアであるプロセスマネージャ410及びネーム空間管理ソフトウェアであるネーム空間マネージャ420を有するものとして見ることができる。プロセスマネージャ410は一般に、メモリ管理と、プロセススケジ

ューリングと、「フォーク」システムコールとの制御のためのプログラムコードからなる。プロセスマネージャ410は周知であり、他で詳細に述べられているので、ここでのプロセスマネージャの説明は、本発明の理解に必要な程度にとどめることとする。

【0096】前に述べたように、核400は、リソースが実際に核400によって実現されるか否かに拘らず、分散形計算環境100内の各リソースへのインタフェースを提供する。例えば、「ブランチ」形の分散形処理システムにおいて、ネーム空間マネージャ420は、図4

(B)に関連して下で述べるような、分散形計算環境100内の各デバイスのインタフェースとして機能するデバイステーブル450(「devtab」)からなる。
【0097】ユーザまたはプロセスのネーム空間内のネームを変換するために、ネーム空間マネージャ420は、図13及び図14に関連して下で述べるバスネームからチャンネルへの変換ソフトウェアである、「バスネーム/チャンネル」コンバータ1200からなる。上で述べたように、ユーザは、特定のファイルを、ユーザのネーム空間内でバスネームを用いて作動させるように指定することができる。「バスネーム/チャンネル」コンバータ1200がバスネームを、作動させるべきファイルを表す特定のチャンネルデータ構造へ翻訳変換する。

【0098】加えて、核400は望ましくは、ファイル樹木の1つのレベルから別のレベルへ移動するために、図15に関連して下で述べる「walk」(ウォーク)(移動)機能1300を実現する。又、図17に関連して下で述べる「domount」(ドゥーマウント)機能1500が望ましくは核400によって供給され、この機能がチャンネルを取って、その意味が、連結され又はマウントされることによって改変されているかどうかを判断し、もし改変されている場合には必要なチャンネル置換を行う。

【0099】遠隔のプロセッサへのプロセスのエクスポート作業中に、図20に示すネーム空間読み出しサブルーチン1700が、望ましくは、マウントテーブル800にアクセスして、ローカルのネーム空間の現在の状態を形成する「連結」コマンド及び「マウント」コマンドの各々に関連する「from」ファイル及び「to」ファイルからなるリストを生成する。

【0100】最後に、核400は望ましくは、ファイルを供給するサーバのネーム空間において、与えられたチャンネルを対応するバスネームに翻訳変換するソフトウェアである、図24に関連して下で述べる「チャンネル/バスネーム」コンバータ2000を供給する。

【0101】図4(B)に示すように、デバイステーブル450は、望ましくは、特定のデバイスによって供給されるファイル上で望むファイル作業(オペレーション)を行うのに利用されるデバイスに特定の手順コールのアレイとして実現される。デバイステーブル450

は、特定のデバイスによって供給されるファイル上で行うことのできる、ファイル開設、読み出し、及び書き込みのような各作業についての手順コールを指すポインタからなる。すなわち、核に常駐する各デバイスは、各基本動作についての手順を含むデバイスドライバによって実現される。

【0102】図4(B)に示すように、デバイステーブル450は、横列461~464のような複数の横列からなり、これらの横列は各々、ディスク装置又はテレタイプ(tty)装置のような、核によって実現された特定のデバイスに連関する。加えて、デバイステーブル450は、縦列452~457のような複数の縦列からなる。デバイステーブル450の縦列452内のエントリは、デバイステーブル450内の与えられた横列に連関する特定のデバイスを表示する。

【0103】縦列453~457のような残りの縦列は各々、特定の手順コールに連関する。例えば横列461に連関するディスク装置によって供給されるファイルを開設するためには核400が、デバイステーブル450のセル468に表示される手順コール「diskopen」(ディスクオープン)を実行する必要がある。

【0104】なお、デバイステーブル450の横列464内のエントリは、図5にマウントデバイス470(又はマウントドライバ)として示す特別な核デバイスに連関する。マウントデバイス(又は核マウントデバイス)470は、核400の一部分で、核400によって実現されない分散形計算環境100内の非核ファイルシステム480~482のようなデバイスの全てにインタフェースを提供する。なお、非核ファイルシステム480~482は、分散形計算環境100内の、核400と同じノード上又は遠隔のノード上に常駐する。

【0105】非核ファイルシステム480~482が核400と同じノードに常駐している場合、プロセス間の通信は望ましくは従来のパイプを用いて行われる。加えて、この非核ファイルシステムが遠隔のノードに位置する場合、プロセス間の通信は1個以上のネットワーク制御を用いて行われる。

【0106】図8に関連して下で更に述べるように、マウントデバイスは、非核ファイルシステムへの、パイプ又はネットワーク接続からなる通信リンク476~478を表す。このリンクは、マウントポイント472~474のような、複数の関連データ構造からなるマウントポイントに対するリンクである。

【0107】なお、マウントデバイス470は望ましくは、ファイルシステム480~482のような接続された非核ファイルシステムの各々に対して1個ずつのマウントポイント472~474を設ける。したがって、与えられた遠隔のファイルシステムにアクセスしようと望むプロセスは、同じマウントポイント472~474を利用することとなる。そのため、マウントデバイス47

0は、連関するサービス機能と通信中の1個以上のプロセスに付いて受信した伝送内容をデマルチプレクス（分離）処理できなければならない。

【0108】分散形計算環境100において、ファイル処理作業（オペレーション）は、しばしば遠隔のファイル上で行われる。概して、遠隔のファイル上で作業を行いたい第1のプロセスは、そのファイルを提供する第2のプロセスへメッセージを送る。送信されたメッセージは、第2のプロセスが、指示されたファイル上で特定のタスクを行うことを要求する。要求されたタスクを第2のプロセスが処理し終ると、第2のプロセスは、単なる肯定応答（アクノリッジ）からなる結果を知らせる応答を第1のプロセスに返す。

【0109】マウントデバイス470は望ましくは、第1及び第2のプロセス間の仲介役として機能し、第1のプロセスによって実行される手順コールをファイル処理作業メッセージに翻訳変換する。このファイル処理作業メッセージは、リンク476～478のような設立された通信リンクを介して第2のプロセスへ送信される。通信リンクを介して送信されるこのファイル処理作業メッセージを、遠隔手順コール（RPC）と称することとする。これについては図8（A）に関連して下で説明する。

【0110】すなわち、もし或るプロセスが、ファイルシステム480から482のような非核ファイルシステムによって提供されるファイル上で特定の処理作業を行いたいと望む場合、このプロセスは、望む処理作業に対応する手順コールを実行することになる。核が、このファイルが非核ファイルシステムによって提供されることを認識した後、マウントデバイス470を利用してこの手順コールを適切な遠隔手順コールに翻訳変換する。

【0111】遠隔手順コールメッセージが遠隔ファイルシステムに、望むタスクの実行を要求する。その後、遠隔のファイルシステムがマウントデバイス470に応答を送る。すると、マウントデバイス470が、ファイル処理作業の結果を要求側のプロセスに送る。このようにして、マウントデバイス470は、要求及び応答のメッセージを手順インタフェースの陰に隠す。

【0112】〔分散形システムにおけるリソースへのアクセス〕或る特定のファイルが或るプロセスによって開設されると、核400が一般に、そのファイルにファイル記述子「fd」又は数字ラベルを割り当てる。加えて、核400は、図7に関連して下で述べる核データ構造（以下、チャンネルとも称する）を利用して、ネーム空間内でプロセスがアクセスする各ファイルを表現する。各400が、図6に示すファイル記述子アレイ500を用いて、割り当てられたファイルにファイル記述子「fd」をマッピングする。

【0113】ファイル記述子アレイ500の各エントリにはファイル記述子「fd」による索引（インデック

ス）が付けられ、ファイル記述子アレイ500の縦列504には、縦列502に表示されるファイル記述子に連関する、チャンネル520～521のような特定のチャンネルを指すポインタを有する。

【0114】一推奨実施例において、各400は、アクティブなプロセスの各々についてファイル記述子アレイ500を維持する。したがって、ファイル記述子アレイ500が指すチャンネル（チャンネルデータ構造）520～522のセットは、特定のプロセスが接続されているファイルの集合、いい替えれば、連関するプロセスのネーム空間内のファイルのセットを表す。

【0115】加えて、各400は望ましくは又、アクティブなプロセスの各々について、図6に示すプロセスデータ構造550を維持する。プロセスデータ構造550は、プロセスの実行を管理するのに必要な情報を既知の仕方で格納する。プロセスデータ構造550は特に、プロセスに連関するファイル記述子アレイ500を指すポインタを格納するためのエントリ555を有する。

【0116】加えて、プロセスデータ構造550は、プロセスのネーム空間内のルートディレクトリ「/」に対応するチャンネル520を指すポインタを格納するためのエントリ560と、プロセスのネーム空間内の現在の作業ディレクトリに対応する、チャンネル521のようなチャンネルを指すポインタを格納するためのエントリ565とを有する。

【0117】前に述べたように、核400は、「チャンネル」と称する核データ構造を利用して、与えられたプロセスがネーム空間内でアクセスする各ファイルを表す。チャンネルデータ構造の適切な実現例を図7に符号600として示す。チャンネルデータ構造（又は、チャンネル）600は望ましくは、ディスク装置又はtty（テレックス）装置のような、対応するファイルを提供する「核サービスデバイスの種類」を記録するための要素605を有する。

【0118】「核サービスデバイスの種類」は、デバイステーブル450の索引として利用され、これにより、適切なデバイスに対応する手順コールのファイル上での実行が可能となる。なお、非核として実現されたファイルについては、「核サービスデバイスの種類」がマウントデバイス470を示すこととなる。

【0119】例えば、もし或る特定のプロセスが或るファイルの読み出しを望む場合、プロセスは、読み出すべきファイルに連関するチャンネルを表示する、ファイル記述子「fd」を含む適切なアーギュメント（探索かぎ）を用いて「read」（読み出し）コマンドを実行する。核400はファイル記述子「fd」を利用して、ファイル記述子アレイ500を介して適切なチャンネル600にアクセスし、その後、データ構造要素605内に表示される、ファイルを提供する「核サービスデバイスの種類」、を特定する。

【0120】その後、この「核サービスデバイスの種類」がデバイステーブル450への索引として用いられる。この特定のデバイスに対応する手順コールが実現される。

【0121】加えて、チャンネル600は望ましくは、チャンネル600に連関するファイルを特定するためにローカルの核400によって割り当てられるファイル識別子(fid)を格納するための要素610と、ファイルを提供するサーバによってサーバのファイル樹木内の各ファイルに割り当てられる独特の番号である「qid」を格納するための要素615とを有する。

【0122】なお、与えられたプロセスは、同じファイルを指し且つ各々が異なるファイル識別子(fid)を持つ複数のチャンネル600を有するけれども、同じファイルを指す各チャンネルは同じ独特の「qid」を有する。「種類」フィールドが、ファイルを提供するリソースを特定するので、「qid」及び「種類」の値が利用されてファイルが独特の仕方で特定され、その結果、2個の異なるチャンネル又はファイル識別子(fid)が同じファイルを指すかどうか判断される。

【0123】チャンネル600は望ましくは又、連関するファイル上で「read」(読み出し)及び「write」(書き込み)のコマンドを実行する際に利用される情報を格納するための1対の要素を有する。要素620が、望ましくは、次の「read」及び「write」のコマンドが開始される連関するファイル内の位置を示す現在の「file offset」(ファイルオフセット)をチャンネルに連関するファイル内に格納する。

【0124】加えて、要素625が、望ましくは、ファイルがアクセスされるモード、いい替えれば、ファイル処理が「read」又は「write」だけでよいか、若しくは両方か、を表示する。

【0125】前に述べたように、もし或るファイルが非核ファイルシステムによって提供される場合、そのファイルは、マウントデバイス470によって提供される、図8(A)に関連して下で述べるマウントポイントを介して間接的にアクセスされる。マウントポイントは、非核ファイルシステムへの通信リンクを表す複数のデータ構造からなる。

【0126】したがって、もし或るチャンネル600が非核ファイルシステムによって提供されるファイルを表す場合、チャンネル600内の要素630は望ましくは、遠隔のファイルサーバのような望む非核ファイルシステムへの通信リンクを表す、図8(A)に示す適切なマウントポイント構造を指すポインタを有する。

【0127】加えて、上に述べたように、ユーザはネーム空間修正コマンドを利用して、少なくとも2個のディレクトリからなる合同ディレクトリをネーム空間内に形成することができる。すなわち、ネーム空間内のチャンネルを合同ディレクトリに等しくすることができ、したが

って合同ディレクトリを表すといえる。チャンネル600が合同ディレクトリを表す場合、チャンネル600内の要素635は望ましくは、合同ディレクトリ内の「現ディレクトリ」を指すポインタを有する。

【0128】図15に関連して下で述べるように、「ウオーク」機能1300を利用して、合同ディレクトリの各要素から要素へ移ることができる。望ましくは、チャンネル600が合同ディレクトリの異なる要素を指すにつれて、ポインタ635が適切に更新される。

【0129】上に述べたように、本発明によれば、与えられたチャンネルに対応するファイルパスネームの正確な再生成を可能にするネーム空間又はパス情報を格納するためのデータ構造が得られる。図12(C)に関連して下で述べる本発明の一態様によれば、各ファイルに連関するパスデータ構造1140が、ファイルのネームと、ネーム空間を通しての特定のファイルへのパスを定めることを可能にする追加情報とを格納する。

【0130】したがって、各チャンネル600は望ましくは、そのチャンネルによって表される与えられたファイルに連関するパスデータ構造1140を指すポインタを格納するための要素640を有する。

【0131】最後に、各チャンネル600は望ましくは、現在特定のチャンネルを指すポインタの数を監視する基準カウンタ645を有する。これにより、基準カウンタが「0」の場合、現在チャンネルを指すポインタがないことを表し、データ構造は、既知の仕方で解放される。

【0132】[非核ファイルシステムとの通信]上に述べたように、或る特定のユーザ又はプロセスのネーム空間内の各ファイルは、そのファイルがローカルの核400によって維持されるか又は非核ファイルシステムによって維持されるかに拘らず、チャンネル600のような、チャンネルデータ構造によって表される。

【0133】しかし、或る特定のチャンネル600が非核で実現されたファイルを表す場合には、そのファイルは、核マウントデバイス470によって与えられる、図5に示すマウントポイント472~474のような、マウントポイントを介して間接的にアクセスされる。なお又、核マウントデバイス470は望ましくは、ファイルシステム480のような接続された各非核ファイルシステムについて、マウントポイント472のような、単一のマウントポイントを形成する。

【0134】そして、形成されたマウントポイント472~474が、連関する非核ファイルシステムによって与えられるファイルへのアクセスを望む複数のプロセスによって利用される。

【0135】(a. 通信リンクの形成)プロセスが最初に、遠隔のファイルサーバのような、図8(A)に示す非核ファイルシステム750に接続するときには、この非核ファイルシステム750(すなわちデバイス)への通信リンク740を設立する必要がある。前に述べたよ

うに、もし非核ファイルシステムが分散形計算環境100内の同じノード上に常駐している場合、通信リンクは望ましくは、従来のパイプとして実現される。しかし、もし非核ファイルシステムが分散形計算環境100内の遠隔のノード上に常駐する場合には、通信リンク740はネットワーク接続が用いられる。

【0136】一般に、オペレーティングシステムは、次のような「パイプ」システムコールを供給する。すなわち、実行されたときに従来のプロセス間通信パイプを形成してパイプの各端を表すチャンネルを特定する2個のファイル記述子、fd、及びfd、を返すコールである。パイプは、普通のファイルのように、適切なファイル記述子fd、又はfd、によってアクセスされる。

【0137】すなわち、非核ファイルシステム750が、第1のファイル記述子fd、を用いて、形成されたパイプの一端から遠隔手順の読み出し及び書き込みを行い、一方、非核ファイルシステム750と通信する1個以上のプロセスが、第2のファイル記述子fd、を利用して、パイプの他端から読み出し及び書き込みを行うことができる。

【0138】しかし、もし非核ファイルシステム750が分散形計算環境100の遠隔のノード上に常駐する場合には、通信リンク740はネットワーク接続が用いられる。一般に、分散形処理システムに連関するオペレーティングシステムは、遠隔のファイルシステムへのネットワーク接続を形成するためのコマンドを出す。

【0139】例えば、推奨する「ブラン9」形の分散形処理システムにおいては、望む宛先の記号的ネーム又はネットワークアドレスのような、適切なアーギュメントを用いて「dial」（ダイヤル）コマンドを実行することによって形成される。

【0140】図10（A）に関連して下で述べるように、「dial」コマンドは、もし必要ならネットワークデータベース900を用いて、記号的ネームをネットワーク宛先へ翻訳変換する。その後、「dial」コマンドは、分散形計算環境100内の指示されたネットワークアドレスへのネットワーク接続の形成を試みる。

【0141】望むネットワーク接続が設立されると、「dial」コマンドは、形成されたネットワーク接続を表すチャンネルデータ構造（又は簡単に、チャンネル）730を特定するファイル記述子「fd」を返す。チャンネル730のようなチャンネルを利用してネットワーク接続上でメッセージの送受信を行う仕方について、図10（B）に関連して下で述べる。

【0142】こうして、図8（A）に示すチャンネル730が、「pipe」又は「dial」コマンドから返されたファイル記述子「fd」によって特定される。チャンネル730は、非核ファイルシステム750と通信するパイプ又はネットワーク接続である通信リンク740を表す。

【0143】一推奨実施例において、特定の、遠隔のファイルシステムと通信するために、予め定められたプロセスが、いま述べた仕方ではパイプ又はネットワーク接続を形成すると、形成された通信リンク740を表すチャンネル730を特定する、返されたファイル記述子が、ローカルのネーム空間内の予め定められた場所にあるファイルに揭示される。

【0144】図11に関連して下で述べる一実施例においては、チャンネル730を特定するファイル記述子「fd」又はポインタを各マウントポイントが格納できるように、特別の核デバイス、「/srv」内にファイルが形成される。

【0145】この仕方では、非核ファイルシステム750のような非核ファイルシステムへの新たな接続を形成する前に、「/srv」ディレクトリ内のファイルを探検して、特定の非核ファイルシステムへの接続が既に設立されているかどうかを判断することができる。もし望む非核ファイルシステムへの接続が前に設立されていることが見出された場合、通信リンクを表すファイル記述子「fd」を「/srv」ディレクトリ内のファイルから取り出し、ネーム空間内の望む場所へマウントすることができる。

【0146】（b. マウントポイントの形成）非核ファイルシステム750への通信リンク740を表すチャンネル730に加えて、核マウントデバイス470（図5）は望ましくは、非核ファイルシステム750に関するアクセス及びパス情報を格納するためにマウントポイントデータ構造（又は簡単に、マウントポイント）710を割り当てる。マウントポイントデータ構造710は望ましくは、非核ファイルシステム750への通信リンク740を表すチャンネル730を指すポインタ（チャンネルデータ構造ポインタ）712を有する。

【0147】加えて、マウントポイントデータ構造710は望ましくは、未決着の遠隔手順コマンドメッセージにそれぞれが連関するタグからなるタグリストを内蔵する待ち行列714を有する。これによって、マウントポイント710が、通信リンク上で受信した応答を適切な要求側プロセスへ送ることが可能となる。

【0148】本発明の一態様によれば、マウントポイントデータ構造710は又望ましくは、図12（A）に関連して下で述べるサーバ樹木バステーブル1100を指すポインタ716を有する。このサーバ樹木バステーブル1100は、非核ファイルシステム750の樹木内のアクセスされたファイルの各々についてのネーム空間情報を内蔵するデータ構造を指すポインタを格納する。下で述べるように、格納されたネーム空間情報により、非核ファイルシステムのファイル樹木を生成することが可能となる。

【0149】一推奨実現例において、チャンネルデータ構造（又は単に、チャンネル）705のような1個のチャネ

ルが、マウントポイント710の形成時に、非核ファイルシステム750によって与えられるファイル樹木のルートディレクトリ「/」を表すように、自動的に割り当てられる。

【0150】上に述べたように、チャンネル705のようなチャンネルが遠隔のファイルを表す場合、チャンネルデータ構造600内の要素630(図7)(すなわち、マウントポイント構造ポイント)には望ましくは、望む遠隔のファイルサーバ750(すなわち、非核ファイルシステム)への通信リンクに対するアクセスを与える適切なマウントポイントデータ構造710を指すポイントが所在する。

【0151】前に述べたように、割り当てられたチャンネル705のような各チャンネルは、チャンネルデータ構造の要素610に表示されるファイル識別子「fid」を有する。各マウントデバイス470は望ましくは、チャンネル705の要素610に表示されるファイル識別子「fid」を含む「attach」(アタッチ)(接続)遠隔手順コールを、通信リンク740を介して非核ファイルシステム750へ送信する。

【0152】「attach」遠隔手順コールは、非核ファイルシステム750によって与えられるファイル樹木のルートにおいて、表示されたファイル識別子「fid」を指すように非核ファイルシステム750に命令する。

【0153】一推奨実施例において、チャンネル705は、非核ファイルシステム750への通信リンク740が維持される間は、非核ファイルシステム750のファイル樹木のルートディレクトリを指す。

【0154】したがって、もしユーザ又はプロセスが、非核ファイルシステム750のファイル樹木内の他のファイルにアクセスしたい場合、既知の仕方で「clone」(クローン)コマンドを用いてチャンネル705をコピーする必要がある。すなわち、マウントテーブル800が、非核ファイルシステム750によって与えられるファイルがアクセスされる都度修正されることにはならない。

【0155】チャンネル705のコピーは最初、非核ファイルシステム750のルートディレクトリ「/」を指す。その後は、図15に関連して下で述べるように、ファイル樹木の或るレベルから別のレベルへ移動する「ウォーク」機能1300を実行することにより、チャンネル705のコピーが非核ファイルシステム750のファイル樹木内の別のファイルを指すように操作することが可能である。これらのコマンドは一般に、遠隔手順コールを用いて非核ファイルシステム750へ送信される。

【0156】(c. 非核ファイルシステムのルートディレクトリのマウント処理) 図3(A)に関連して上に述べたように、非核ファイルシステム750のファイル樹木は、非核ファイルシステム750のルートディレクト

リを表すチャンネル705をローカルのネーム空間内に既に所在するファイルに等しくすることによって、ローカルのネーム空間内に位置させることができる。

【0157】いい替えれば、「マウント」コマンドを実行することにより、ネーム空間内の、チャンネルによって表されるファイル上にチャンネル705をマウントすることができる。例えば、もし図8(A)に示す通信リンク740が、図3(A)に示すユーザ端末305から遠隔のファイルサーバ310への接続を表す場合、チャンネル705はこの遠隔のファイルサーバ310のネーム空間325のルートディレクトリ「/」を表す。

【0158】したがって、遠隔のファイルサーバ310のファイル樹木をユーザ端末305に連関するネーム空間320内に位置させるためには、チャンネル705をユーザ端末のネーム空間320内のファイルX上にマウントする必要がある。

【0159】前に述べたように、核400は、図8(B)及び図9に示すマウントテーブル800及び関連するデータ構造を利用して、ローカルのネーム空間を形成するためにユーザ又はプロセスが実行する一連の「連結」及び「マウント」のコマンドを格納する。

【0160】したがって、図3(A)及び図3(B)の例に関連して引続き説明すると、もしユーザ端末305に連関してネーム空間320'上で処理作業を行うユーザがファイル「/X/D」を開こうと試みる場合、核400は、マウントテーブル800を利用して、ファイルX及び階層のファイルXよりも下方のファイルにアクセスする要求を、ファイルサーバ310のファイル樹木内の、ファイルDのようなファイルへの参照として解釈する。

【0161】一推奨実施例において、核400は、ファイルがバスネームを用いてアクセスされる都度、図13及び図14に関連して下で述べる「バスネーム/チャンネル」コンバータ1200を実行する。これによって特に、「連結」又は「マウント」コマンドからチャンネル置換が必要かどうか判断される。

【0162】なお、「連結」又は「マウント」コマンドのアーギュメントは、「from」チャンネル及び「to」チャンネルである。したがって、マウントテーブル800の各エントリは、「from」エントリと「to」エントリとの対からなる。もし或るチャンネルがマウントテーブル800において「from」チャンネルに見える場合、別のチャンネルがこのチャンネル上にマウントされており、チャンネル置換を行う必要がある。

【0163】したがって、バスネームを用いてファイルがアクセスされる都度、各400は、バスネームをこれに対応するチャンネルに変換し、マウントテーブル800について探索を行ってその特定のチャンネルが「他からマウントを受けて」いるかどうか、いい替えれば、そのチャンネルがマウント「from」チャンネルのリストに現れ

るかどうかを判断する。もしチャンネルがマウント「from」チャンネルのリストに見出された場合、マウントテーブル800内のこれに対応するマウント「to」エントリが取り出され、必要なチャンネル置換が行われる。

【0164】上に述べたように、チャンネル600のような各チャンネルの要素615及び605にそれぞれ格納されている「qid」及び「type」の値が、そのチャンネルに対応するファイルを独特の仕方と特定する。こうして、チャンネルが望ましくは、その「qid」及び「type」の値によってマウントテーブル内で特定される。一推奨実施例において、マウントテーブル800内のエントリが、「他からマウントを受けた」チャンネルの「qid」値に基づくハッシュ値によって索引が付けられる。

【0165】したがって、或る特定のチャンネルが「他からマウントを受けて」いるかどうかを判断するには、マウントテーブル800に、対応する「qid」値を入れる。その後、マウント「from」チャンネルのリストにおいて、同じ「qid」及び「type」の値を有するチャンネルが存在するかどうかの探索が行われる。もし一致が見出された場合、核400が、対応するマウント「to」チャンネルエントリによって表示されるチャンネルについて置換を実行する。

【0166】概念的には、マウントテーブル800を、1対の縦列805及び810からなる最上位レベルのデータ構造として見ることができる。ハッシュアドレス縦列805内の各エントリには、独特のハッシュ値が入り、これが横列815、820、及び825のようなマウントテーブル800の各横列についての索引又はアドレスとして機能する。

【0167】なお、いくつもの異なる「qid」値が同一のハッシュ値に合わせて調整される。したがって、横列820のようなマウントテーブル800の各横列が、各々「qid」及び「type」の値によって独特の仕方と特定された複数の「他からマウントを受けた」チャンネルに関連する。このため、「リンクされたリストを指すポインタ」の縦列810は、もしエントリがあるとすれば、望ましくはハッシュ値に関連する、データ構造830、832、及び837からなるリンクされたデータ構造リスト826を指すポインタ828を有する。

【0168】マウント「from」及びマウント「to」データ構造についての現在推奨の一実施例を図9に示す。概して、図8(B)に示すように、構造830、832、及び837のような構造についてのリンクされたマウント「from」データ構造リスト826内の各データ構造は、リンクされたリスト内の次のデータ構造を指すポインタ829のようなポインタを有する。リンクされたリスト内の最後のデータ構造837は望ましくは図8(B)に示すような「ゼロ」ポインタを有する。

【0169】前に述べたように、マウントテーブル80

0内の各エントリは、「他からマウントを受けた」チャンネルを表す「from」データ構造と、1個以上のマウントされたチャンネルを表す少なくとも1個の「to」データ構造とからなる。図9に示すように、最上位レベルマウントテーブル800からのリンクされたリストのポインタ828が、リンクされたデータ構造リスト826内のマウント「from」データ構造830を指す。

【0170】リンクされたデータ構造リスト826の一部をなすマウント「from」リンクリストデータ構造830のような、マウント「from」データ構造は望ましくは、「他からマウントを受けた」チャンネルを指すポインタを格納するためのエントリ840と、同じく連関するマウント「to」リンクリストデータ構造850を指すポインタを格納するためのエントリ845とを有する。

【0171】なお、もしマウントされたチャンネルが合同ディレクトリの一部である場合、マウント「to」データ構造は、リンクされたリスト880のようなリンクされたデータ構造リストの形を取る。加えて、データ構造830のようなマウント「from」データ構造の各々は望ましくは、リンクされたリスト826内の次の要素を指すポインタ829を格納するためのエントリ848を有する。

【0172】一推奨実施例において、核400は「ドゥーマウント」機能1500を有し、この機能がチャンネルを取って、その意味が、「他からマウントを受ける」ことによって改変されているかどうかを判断し、もし改変されている場合には必要なチャンネル置換を行う。

【0173】下で述べるように、「ドゥーマウント」機能1500は、受信されたチャンネルから「qid」を取り出し、ハッシュ値を生成し、マウントテーブル800の縦列810内の適切なエントリからリンクリストポインタ828を利用して、リンクされたデータ構造リスト（リンクリスト）826のような適切な、マウント「from」データ構造のリンクされたリストにアクセスする。

【0174】その後、核400が、リンクされたデータ構造リスト826内の、データ構造830のようなマウント「from」データ構造の各々を指すポインタに逐次従って、もしリンクリスト内の或るマウント「from」データ構造によって指されるチャンネルが、「ドゥーマウント」機能1500へ送られたチャンネルと同じ「qid」及び「type」の値を有するかどうかを判断する。

【0175】もし或るチャンネルが、一致する「qid」及び「type」の値を有することが判った場合、マウント「from」データ構造830のエントリ845内のポインタに従って、データ構造850のようなマウント「to」データ構造の、連関するリンクされたデータ構造リスト880の最初の要素に到達する。望ましく

は、各マウント「to」データ構造850が、マウントされたチャンネルを指すポインタを格納するための要素855を有する。

【0176】加えて、このマウント「to」データ構造850は望ましくは、マウントテーブル800内のそのエントリを独特の仕方にて特定する「mountid」値を格納するための要素860を有する。一推奨実施例において、エントリが付加される都度、「mountid」値が増値され、これにより、この値がタイムスタンプとして機能し、マウントテーブルのエントリを、対応する「連結」及び「マウント」のコマンドが実行された順に取り出すことが可能となる。

【0177】加えて更に、マウント「to」のリンクされたリストのデータ構造850は望ましくは、連関するマウント「from」のリンクされたリストのデータ構造830に戻ってこれを指すポインタを格納する要素865を有する。

【0178】加えて、マウント「to」のリンクされたリストのデータ構造850は、マウントテーブルエントリに連関する「連結」又は「マウント」のコマンドに指定されたマウント順序フラッグを格納するための要素870を有する。いい替えれば、連関する「連結」又は「マウント」のコマンドが、合同ディレクトリ内の「現ディレクトリ」を合同ディレクトリ内の他のディレクトリの前にすべきと表示していたか又は後にすべきと表示していたかである。

【0179】最後に、マウント「to」のリンクされたリストのデータ構造850は望ましくは、マウント「to」データ構造のリンクされたリスト880内の次の要素を指すポインタを格納するためのエントリ875を有する。

【0180】[ネットワークの表現]前に述べたように、図10(A)に示すネットワークデータベース900を利用して記号的マシンネームとネットワーク宛先アドレスとの間の翻訳変換が行われる。例えば、ネットワーク接続を設立する際、ユーザ又はプロセスが、記号的マシンネームによって特定される望む遠隔のマシンへの接続を要求する。すると、接続ステップの間に、表示された記号的マシンネームを対応するネットワーク宛先アドレスに翻訳変換するためにネットワークデータベース900がアクセスされる。

【0181】加えて、下に述べる本発明の態様によれば、ネーム空間を再形成するときにネットワーク宛先アドレスを再び翻訳変換して記号的マシンネームに戻すことがしばしば必要となる。

【0182】したがって、図10(A)に示すネットワークデータベース900は望ましくは、1対の縦列925及び930からなる。データベース900の、横列905、910、及び915のような各横列は望ましくは、それぞれ或る特定の遠隔のマシンと連関関係にある。

る。

【0183】縦列925内の各エントリが、特定のマシンに連関する記号的マシンネームを表示する。縦列930内の各エントリは望ましくは、特定のマシンが接続されるネットワークと適切なネットワーク宛先アドレスとのリストを有する。例えば、マシン「helix」は「ip」ネットワーク上に宛先アドレス135.104.9.31を有する。

【0184】この仕方では、もしユーザ又はプロセスが、マシン「helix」のような特定の遠隔のマシンへのネットワーク接続を要求した場合、核400がネットワークデータベース内の適切なエントリにアクセスして、そのマシンの、対応するネットワーク宛先アドレスを取り出す。望ましくは、接続の設立が成功するまで、取り出された各宛先アドレスへの接続の試みを反復して行う。

【0185】ネットワークは、分散形計算環境100内の他のリソースと同様に、望ましくは階層形ファイルシステムとして表現される。図10(B)に、ネットワークに連関するネーム空間950の一部を例示する。ネーム空間950は望ましくは、ディレクトリ「/net」を有し、ディレクトリ「/net」は、ネットワーク「il」及び「tcp」のようなアクセスされる各ネットワークについてのサブディレクトリである。

【0186】加えて、ネットワークディレクトリ「/net/il」のような各ネットワークディレクトリは望ましくは、「clone」(クローン)ファイルと、番号を付けたディレクトリのセットとを有する。番号を付けたディレクトリの各々は、連関する接続の制御と同接続上での通信のためのファイルを有する。

【0187】下で述べるように、新たなネットワーク接続は、ファイル「/net/il/clone」のような、望むネットワークのネットワークディレクトリ内の「clone」ファイルを開いて未使用の接続を予約することによって割り当てられる。開設コマンドによって返されたファイル記述子「fd」によって、新たな接続の「制御」ファイルを指すチャンネルが特定される。

【0188】その後、接続を設立するために、「connect 135.104.9.30!1708」のようなASCII(アスキー)アドレス文字列が新たな接続の「制御」ファイルに書き込まれて、「il」ネットワーク上の「bootes」マシンへの接続が行われる。

【0189】その後、連関するデータファイルに対する読み出し及び書き込みにより、設立された接続上での受信及び送信がそれぞれ可能となる。なお、「listen」(聴取)ファイルを用いてネットワークからの着信コールが受け入れられる。又、「local」(ローカル)ファイルは一般に、アドレッシング情報を有し、「remote」(遠隔)及び「status」

(状態)のファイルは接続状態についての情報を出す。
 【0190】「プラン9」形の分散形計算システムのような分散形計算環境100でのネットワーク接続の形成及び利用の更に詳細な内容については、文献(David Presotto & Philip Winterbottom, "The Organization of Networks in Plan 9", Proc. of the Winter 1993 US ENIX Technical Conference, San Diego, CA., pp. 43-50 (1993))を参照されたい。ここに本文献を本出願の参照文献とする。

【0191】図8(A)に関連して上に述べたように、パイプ又はネットワーク接続のような通信リンクが非核ファイルシステムに対して設立される都度、形成された通信リンクを表すチャンネルを特定するために「pipe」又は「dial」コマンドによって返されるファイル記述子「fd」が望ましくは、ローカルのネーム空間内の予め定められた場所にあるファイルに揭示される。

【0192】図11に示す一実施例において、形成されたマウントポイントの各々についてディレクトリ「/srv」内にファイルが形成され、その通信リンクを表すチャンネルを特定するファイル記述子「fd」又はポインタが格納される。

【0193】例示の実施例において、パイプに対応するチャンネルを指すポインタを格納するために「/srv」ディレクトリ内に形成されたファイルには、次の構成に基づいてファイル名が付けられる。すなわち、programname.username.procid(プログラム名.ユーザ名.プロセスid)の構成である。したがって、ファイル「8-1/2.philw.35」には、ユーザ「philw」に対する「プラン9」形ウィンドウシステム「8-1/2」へのパイプに関連するチャンネルを指すポインタが格納される。

【0194】同様に、ネットワーク接続に対応するチャンネルを指すポインタを格納するために「/srv」ディレクトリ内に形成されたファイルには、次の構成に基づいてファイル名が付けられる。すなわち、network!machine name(ネットワーク!マシン名)の構成である。したがって、ファイル「il!bootes」には、「il」ネットワークを介してのマシン「bootes」へのネットワーク接続に関連するチャンネルを指すポインタが格納される。

【0195】この仕方では、デバイス750のような非核ファイルシステムへの新たな接続を形成する前に、「/srv」ディレクトリ内のファイルについて探索を行ってその特定の非核ファイルシステムへの接続が既に設立されているかどうかを判断することができる。そして、もし望む非核ファイルシステムへの接続が既に設立されている場合、「/srv」ディレクトリ内に表示されているチャンネルをネーム空間内の望む場所にマウントすることができる。

【0196】加えて、「/srv」ディレクトリ内のフ

ファイルに格納されている情報を利用して、図18に関連して下で述べるように処理タスクのエクスポート作業中にcpuサーバへの送信用にネーム空間を一体のパッケージにまとめる(パッケージ化する)ときにパイプ及びネットワーク接続を特定することができる。

【0197】[ネーム空間情報データ構造]前に述べたように、ネーム空間を形成するためにユーザ又はプロセスによって実行された「マウント」及び「連結」のコマンドのリストが、望ましくはこれらのコマンドが実行された順にマウントテーブル800から取り出される。しかしここで注記したいのは、マウントテーブル800から取り出された「マウント」及び「連結」のコマンドのリストが、バインド(連結)され又はマウント(搭載)されたその特定のチャンネルデータ構造によって表され、ファイルバスネームでは表されないことである。

【0198】このため、本発明の別の態様により、図12(A)~図12(C)に示す、バスデータ構造が、各チャンネルに対応するバスネームを定めることを可能にするバス情報を格納する。加えて、下で述べるように、格納されたバス情報により、接続された非核ファイルシステムの各々についての樹木内のアクセスされたファイルのネーム空間を再形成することが可能になる。

【0199】前に述べたように、図8(A)に関連して上に述べた、遠隔のサーバへのアクセスを与えるマウントポイントデータ構造710は、図12(A)に示すサーバ樹木バステーブル1100を指すポインタ716を有する。

【0200】サーバ樹木バステーブル1100は、マウントポイント710に関連する非核ファイルシステム750の階層形ファイル樹木内の、アクセスされたファイルの各々についてのバス情報を含むデータ構造を指すポインタを格納する。格納されたバス情報によって、非核ファイルシステムのファイル樹木を下で述べる仕方では生成することが可能となる。

【0201】図12(A)に示すように、サーバ樹木バステーブル1100は望ましくは、遠隔のファイルシステム750のルートディレクトリを表すチャンネル705のような、特定の非核ファイルシステムのルートディレクトリを表すチャンネルに関連するバスデータ構造を指すポインタ1102からなる。

【0202】下で述べるように、非核ファイルシステムのルートのバスデータ構造を指すポインタ1102を利用して、ルートディレクトリまでの全行程を戻って階層内の与えられたファイルからバスネームが、いつ再構築されたかを特定することができる。

【0203】加えて、サーバ樹木バステーブル1100は望ましくは、図12(B)に示すサーバ樹木バスリスト1110を指すポインタ1104を有する。このサーバ樹木バスリスト1110は望ましくは、関連する非核ファイルシステムの全てについてのバス情報を有するデ

ータ構造を指すポインタのリストである。

【0204】一推奨実施例において、サーバ樹木バスリスト1110内のエントリは、与えられたファイルについてのバスネーム情報に基づく。いい替えれば、一推奨実施例において、サーバ樹木バスリスト1110内のエントリは、与えられたファイルの「バスネーム」に基づくハッシュ値によって索引が付けられる。

【0205】この仕方では、与えられたファイルに関連する、データ構造1140のような適切なバスデータ構造を、例えば「ウオーク」機能1300の実行中、バスネームに基づいてアクセスされるファイルについて特定することができる。なお、もし或る特定のチャンネルについてのバス情報を望む場合、チャンネルデータ構造600 (図7) からバス構造ポインタ640を取り出すことによって、適切なデータ構造に直接アクセスすることができる。

【0206】したがって、バスネームによって特定された、或る特定のファイルについてのバス情報にアクセスするためには、与えられたファイルのバスネームに基づくハッシュ値をサーバ樹木バスリスト1110に入れる。

【0207】概念的には、サーバ樹木バスリスト1110を、1対の縦列1122及び1124からなる最上位レベルのデータ構造として見ることができる。ハッシュアドレス縦列1122内の各エントリには、独特のハッシュ値が入り、これが横列1115、1118、及び1120のようなサーバ樹木バスリスト1110の各横列についての索引又はアドレスとして機能する。

【0208】なお、いくつもの異なる「pathname」(バスネーム) 値が同一のハッシュ値に合わせて調整される。したがって、横列1118のようなサーバ樹木バスリスト1110の各横列が、複数のバスデータ構造に関連する。このため、「リンクされたリストを指すポインタ」の縦列1124は、もしエントリがあるとなれば、望ましくはハッシュ値に関連する、データ構造1130、1132、及び1137からなるリンクされたデータ構造リスト1126を指すポインタ1128を有する。

【0209】バスデータ構造11140についての現在推奨の一実施例を図12(C)に示す。概して、図12(B)に示すように、構造1130、1132、及び1137のようなリンクされたバスデータ構造リスト1126内の各データ構造は、リンクされたリスト内の次のデータ構造を指すポインタ1129のようなポインタを有する。リンクされたリスト内の最後のデータ構造1137は望ましくは図12(B)に示すような「ゼロ」ポインタを有する。

【0210】リンクリストバスデータ構造1140のような各バスデータ構造は望ましくは、連関するファイルのネームを格納するためのエントリ1142と、現ファ

イルの親のファイルバスデータ構造を指すポインタを格納するためのエントリ1144と、基準カウンタ1146と、リンクされたリスト内の次の要素を指すリンクされたリストのポインタ1148とを有する。

【0211】図24に関連して下で述べるように、「チャンネル/バスネーム」コンバータ2000は、データ構造1140のようなバスデータ構造を利用して、チャンネルとして表される、与えられたファイルについてのバスネームを構築する。概して、チャンネルが与えられると、「チャンネル/バスネーム」コンバータ2000が、表示されたファイルについてバスデータ構造1140にアクセスし、エントリ1142からファイルネームを取り出す。

【0212】それから、エントリ1144から取り出された親のファイルのバスデータ構造を指す一連のポインタに順次従って、ルートディレクトリに到達するまでファイル樹木を登りバスネームを構築する。

【0213】[ネーム空間プロセス]前に述べたように、核400は、ユーザまたはプロセスのネーム空間内の与えられたバスネームを、表示されたファイルを表す適切なチャンネルに翻訳変換するための「バスネーム/チャンネル」コンバータ1200を提供する。上に述べたように、ユーザまたはプロセスは、ユーザまたはプロセスのネーム空間内のバスネームによって特定されるいくつもの異なるファイル作業を1つのファイル上で行うことができる。バスネームアークメントを受信して作業場所のファイルを特定するこれらのファイル作業の各々は望ましくは、「バスネーム/チャンネル」コンバータ1200を呼んで、表示されたバスネームを、その作業場所のファイルを表す特定のチャンネルデータ構造へ翻訳変換する。

【0214】下で述べるように、「バスネーム/チャンネル」コンバータ1200の処理プロセスは、3個の主要部分から構成される。第1の部分及び第2の部分は、表示されたファイルを含むディレクトリを表す適切なチャンネルを特定する。なお、「バスネーム/チャンネル」コンバータ1200を呼ぶこれらの異なるファイル作業においては、アクセスモードによって特定されて、その表示されたファイル上で、異なるタスクを行うことになる。

【0215】「open」(開設)又は「create」(形成)のような大抵のファイル作業については、マウントテーブル800に表示される全てのチャンネル置換を行うことが望ましい。しかし、ファイル作業が「作業ディレクトリの設定」又は「マウント」コマンドである場合、そのファイル作業は、合同ディレクトリ自体を表すチャンネル上で行うべきであり、合同ディレクトリ内の第1の要素を表すチャンネル上で行うべきではない。

【0216】この仕方では、もし或るチャンネルが次の「連結」又は「マウント」コマンドによって合同ディレクトリ内の第1の要素の前に挿入された場合、その新しい第

1の要素が、合同ディレクトリを表すチャンネルによって指されることになる。そのため、「バスネーム／チャンネル」コンバータ1200の処理プロセスの第3の部分1245が、受信されたアクセスモードアークメントを点検評価し、置換を行わないのが適切な場合には、最後のバスネーム要素についてマウントテーブル800に表示されているチャンネル置換を行わない。

【0217】図13に示すように、ステップ1204において、「バスネーム／チャンネル」コンバータ1200がプロセスに入り、表示されたバスネームアークメント（探索かぎ）を用いて処理を開始する。ステップ1208～1224に対応する「バスネーム／チャンネル」コンバータ1200の処理プロセスの第1の部分1202の間、バスネームの第1の要素が分析される。

【0218】なお、バスネームは一般に、そのバスネームについての異なる発起点をそれぞれが特定するような3個の文字のうちの1個で始まる。そして、ステップ1208において、バスネームのアークメント中の第1の文字が、そのバスネームがルートディレクトリに関して指定されていることを表示する斜線文字かどうか点検される。

【0219】もしステップ1208において、バスネームのアークメント中の第1の文字がルートディレクトリ「/」を表示すると判断された場合、ステップ1212において、ルートディレクトリを表すチャンネルを指すポインタがプロセスデータ構造550（図6）の要素560から取り出され、対応するチャンネルがクローン処理すなわちコピーされる。

【0220】しかし、もしステップ1208において、バスネームのアークメント中の第1の文字がルートディレクトリ「/」を表示しないと判断された場合には、ステップ1216において、バスネームのアークメント中の第1の文字が、そのバスネームが現作業ディレクトリに関して指定されていることを表示するドット文字「.」かどうか点検される。

【0221】もしステップ1216において、バスネームのアークメント中の第1の文字が現作業ディレクトリ「.」を表示すると判断された場合、ステップ1220において、現作業ディレクトリを表すチャンネルを指すポインタがプロセスデータ構造550（図6）の要素565から取り出され、対応するチャンネルがクローン処理すなわちコピーされる。

【0222】しかし、もしステップ1216において、バスネームのアークメント中の第1の文字が現作業ディレクトリ「.」を表示しないと判断された場合には、バスネームのアークメント中の第1の文字は、そのバスネームが核デバイスに関して指定されていることを表示するハッシュ文字「#」に違いない。

【0223】したがって、ステップ1224において、プロセスデバイスを表示する「#p」のようなバスネー

ムの第2の文字によって特定される核デバイスが取り付けられる。「attach」（取り付け）コマンドによって既知の仕方でも返されたファイル記述子「fd」が変数「channel」（チャンネル）に割り当てられる。

【0224】「バスネーム／チャンネル」コンバータ1200の処理プロセスの第1の部分1202の実行の後、変数「チャンネル」には、バスネームのアークメント中の第1の文字（すなわち要素）によって表示されたディレクトリ又はファイルを表すチャンネルが含まれる。ステップ1228～1240に対応する「バスネーム／チャンネル」コンバータ1200の処理プロセスの第2の部分1226の間、バスネームアークメントの各要素が、最後の要素に到達するまで逐次、点検処理される。

【0225】ステップ1228において、バスネームアークメントの「次の要素」が取り出される。その後、ステップ1232において、図15に関連して下で述べる「walk」（ウォーク）機能が、変数「チャンネル」の現在値と前のステップにおいて取り出された「次の要素」とを用いて実行される。「ウォーク」機能1300の結果が変数「チャンネル」に割り当てられる。

【0226】概して、「ウォーク」機能1300は、バスネームにおいて「次の要素」によって表示されるファイル又はディレクトリを求めて、「チャンネル」の現在値によって表されるディレクトリ内で探索を行う。

【0227】ステップ1232における「ウォーク」機能の実行に続いて、ステップ1236において、図17に関連して下で述べる「domount」（ドゥーマウント）機能1500が実行される。概して、「ドゥーマウント」機能1500は、マウントテーブル800にアクセスして、変数「チャンネル」の現在値によって表されるチャンネルが「他からマウントを受けて」いるかどうかを判断する。

【0228】もし変数「チャンネル」の現在値に対応するチャンネルが「他からマウントを受けて」と判断された場合には、変数「チャンネル」の値が「ドゥーマウント」機能1500によって、マウントされたチャンネルが表示されるように翻訳変換される。

【0229】その後、ステップ1240において、最後の要素以外に点検評価すべきバスネームアークメント内の要素がまだあるかどうかを判断する点検が行われる。もしステップ1240において、点検評価すべきバスネームアークメント内の要素がまだあると判断された場合、プロセスはステップ1228に戻って、上に述べた仕方でも残りのバスネーム要素の処理を継続する。

【0230】しかし、もしステップ1240において、点検評価すべきバスネームアークメント内の要素がもはやないと判断された場合、プロセスは図14のステップ1250に進む。「バスネーム／チャンネル」コンバータ1200の処理プロセスの第2の部分1226の実行の後、変数「チャンネル」は、表示されたバスネーム内の

最後の要素の次を、いい替えれば、パスネームアークメントによって特定されたファイル（又はディレクトリ）を指す。を指している。

【0231】前に述べたように、「パスネーム／チャンネル」コンバータ1200によって受信されたアークメントは、現ファイル作業のアクセスモードを特定する。上に述べたように、「アクセス」モードが、マウントテーブル800に表示されたチャンネル置換がパスネームの最後の要素に対して行われたかどうかを判断する。

【0232】ステップ1250～1290に対応する「パスネーム／チャンネル」コンバータ1200の処理プロセスの第3の部分1245がアクセスモードのアークメントを点検評価して、そのファイルを表す適切なチャンネルを特定する。

【0233】ステップ1250において、現アクセスモードが「アクセス」かどうかを判断する点検が行われる。もしステップ1250において、現アクセスモードが「アクセス」であると判断された場合、ステップ1254において、図15に示す「ウオーク」機能1300が変数「チャンネル」の現在値と、パスネームの「最後の要素」とを用いて実行される。「ウオーク」機能の結果が変数「チャンネル」に割り当てられる。

【0234】「ウオーク」機能1300は、パスネームにおいて「最後の要素」によって表示されるファイル又はディレクトリを求めて、「チャンネル」の現在値によって表されるディレクトリ内で探索を行う。

【0235】ステップ1254における「ウオーク」機能の実行に続いて、ステップ1258において、図17に示す「domount」（ドゥーマウント）機能1500が実行される。

【0236】「ドゥーマウント」機能1500は、マウントテーブル800にアクセスして、変数「チャンネル」の現在値によって表されるチャンネルが「他からマウントを受けて」いるかどうかを判断し、もしそうなら、マウントされたチャンネルを表示するために、変数「チャンネル」の値が「ドゥーマウント」機能1500によって翻訳変換される。その後、プロセスは、ステップ1290において、作業場所のファイルを今や表している変数「チャンネル」の現在値を持って呼び出し側のファイル作業に戻る。

【0237】しかし、もしもステップ1250において、現アクセスモードが「アクセス」に等しくないと判断された場合、ステップ1262において、現アクセスモードが「作業ディレクトリの設定」又は「マウント」に等しいかどうかを判断するための点検が行われる。

【0238】もしステップ1262において、現アクセスモードが「作業ディレクトリの設定」又は「マウント」に等しいと判断された場合、チャンネルが合同ディレクトリ自体を表すのが望ましく、合同ディレクトリ内の第1の要素を表すのは望ましくない。

【0239】それで、ステップ1264において、図15に示す「ウオーク」機能1300が、変数「チャンネル」の現在値とパスネームの「最後の要素」とを用いて実行され、この実行により、パスネームにおいて「最後の要素」によって表示されるファイル又はディレクトリを求めて、「チャンネル」の現在値によって表されるディレクトリ内で探索が行われる。「ウオーク」機能1300の結果は、変数「チャンネル」に割り当てられる。

【0240】その後、プロセスは、ステップ1290において、「ドゥーマウント」機能1500を呼び出すことなく、変数「チャンネル」の現在値を持って呼び出し側のファイル作業に戻る。

【0241】しかし、もしステップ1262において、現アクセスモードが「作業ディレクトリの設定」又は「マウント」に等しくないと判断された場合には、ステップ1268において、現アクセスモードが「開設」又は「形成」に等しいかどうかを判断するための点検が行われる。

【0242】もしステップ1268において、現アクセスモードが「開設」又は「形成」に等しいと判断された場合、ステップ1272において、図15に示す「ウオーク」機能1300が、変数「チャンネル」の現在値とパスネームの「最後の要素」とを用いて実行され、この実行により、パスネームにおいて「最後の要素」によって表示されるファイル又はディレクトリを求めて、「チャンネル」の現在値によって表されるディレクトリ内で探索が行われる。「ウオーク」機能1300の結果が変数「チャンネル」に割り当てられる。

【0243】ステップ1272における「ウオーク」機能の実行に続いて、ステップ1276において、図17に示す「domount」（ドゥーマウント）機能1500が実行される。

【0244】「ドゥーマウント」機能1500は、マウントテーブル800にアクセスして、変数「チャンネル」の現在値によって表されるチャンネルが「他からマウントを受けて」いるかどうかを判断し、もしそうなら、マウントされたチャンネルを表示するために、変数「チャンネル」の値が「ドゥーマウント」機能1500によって翻訳変換される。

【0245】なお、ファイル開設のタスクは、ある種の許可について検証を受ける必要がある。したがって、ステップ1280において、ファイルがアクセスされる仕方が点検評価される。

【0246】その後プロセスはステップ1290において、作業場所のファイルを今や表している変数「チャンネル」の現在値を持って呼び出し側のファイル作業に戻る。

【0247】しかし、もしもステップ1268において、現アクセスモードが「開設」又は「形成」に等しくないと判断された場合には、有効なアクセスモードが全

て試みられたわけで、現アクセスモードはエラーに違いない。したがって、望ましくは、プロセスが呼び出し側のファイル作業に戻る前に、ステップ1284においてエラーフラグがセットされる。

【0248】前に述べたように、図15に示す「ウオーク」機能1300は、「チャンネル」とバスネーム「要素」とをアーギュメントとして受信する。下で述べるように、「ウオーク」機能1300は、バスネーム「要素」によって表示されるファイル又はディレクトリを求めて、「チャンネル」の現在値によって表されるディレ

トリ内で探索を行う。
【0249】「ウオーク」機能1300は、図16に示すデバイスウオークサブルーチンを利用して、その特定のディレクトリが、表示されたファイルを有するかどうかを実際に判断する。加えて、受信されたチャンネルが、「ウオーク」成功時に新たなファイルを指すので、デバイスウオークサブルーチン1400は又、要すれば、適切なバスデータ構造1140（図12（C））を更新する。

【0250】図15に示すように、「ウオーク」機能1300は望ましくは、ステップ1305においてプロセスに入り、処理を開始する。「ウオーク」機能1300のアーギュメントは、或る特定のファイルを求める探索が行われるディレクトリを表す「チャンネル」であって、与えられたネーム要素によって表示される「チャンネル」、からなる。

【0251】ステップ1310において、チャンネル600のような表示されたチャンネルの核サービスデバイス「種類」要素605が点検評価され、これによりそのファイルを提供する核デバイスが特定される。前に述べたように、核サービスデバイス「種類」は、デバイステーブル450において索引として利用される。そして、「ウオーク」作業に対応するデバイステーブル450の縦列内のエントリによって表示されるような、適切なデバイスウオークサブルーチンがステップ1315において実行される。

【0252】例示のデバイスウオークサブルーチン1400について、図16に関連して下で述べる。前に述べたように、デバイスウオークサブルーチン1400は、表示されたディレクトリを表すチャンネルが、受信されたバスネーム「要素」によって特定されるファイルを有するかどうかを判断し、もし成功の場合は又、適切なバスデータ構造1140を更新する。

【0253】その後、ステップ1320において、デバイスウオークサブルーチン1400が成功だったかどうかを判断する点検が行われる。いい替えれば、ネーム「要素」アーギュメントによって表示されるファイルが、「チャンネル」アーギュメントによって表されるディレクトリ内に見出されたかどうかである。もしステップ1320において、デバイスウオークサブルーチン14

00が成功だったと判断された場合、受信されたバスネーム「要素」によって表示されるファイルを今や指すような変数「チャンネル」の現在値が返される。

【0254】しかし、もしステップ1320において、デバイスウオークサブルーチン1400が不成功だったと判断された場合には、受信されたチャンネルは、そのファイルを含む合同ディレクトリを表す。したがって、ステップ1325において、受信された「チャンネル」が合同ディレクトリを表すかどうかを判断する点検が行われる。

【0255】もしステップ1325において、「チャンネル」が合同ディレクトリを表すと判断された場合、マウント合同ポインタが、チャンネルデータ構造（チャンネル）600のような受信された「チャンネル」の要素635から取り出される。前に述べたように、チャンネル600のようなチャンネルが合同ディレクトリを表すとき、マウント合同ポインタ635が、合同ディレクトリ内の現ディレクトリを表すチャンネルを指す。

【0256】その後、チャンネル600のような、マウント合同ポインタ635によって指されたチャンネルの核サービスデバイス「種類」要素605が、ステップ1332において点検されて、ディレクトリを供給し且つデバイステーブル450において索引として利用される核サービスデバイス「種類」が特定される。そして、ステップ1334において、適切なデバイスウオークサブルーチンが実行される。

【0257】ステップ1336において、ステップ1334で実行されたデバイスウオークサブルーチンが成功だったかどうかを判断するための点検が行われる。いい替えれば、ネーム「要素」アーギュメントによって表示されるファイルが、マウント合同ポインタ635によって指される現ディレクトリ内に見出されたかどうか、である。

【0258】もしステップ1336において、ステップ1334で実行されたデバイスウオークサブルーチン1400が成功だったと判断された場合、バスネーム「要素」によって表示されるファイルを今や指すような変数「チャンネル」の現在値がステップ1350において返される。

【0259】しかし、もしステップ1336において、ステップ1334で実行されたデバイスウオークサブルーチン1400が不成功だったと判断された場合には、他のディレクトリがもしあれば、それらのディレクトリが点検評価される。

【0260】それから、ステップ1338において、点検評価されている現チャンネルが、合同ディレクトリの「最後の要素」を表すかどうか、いい替えれば、マウント「to」データ構造850の要素875から取り出された、リンクされたリスト880の「次の要素」を指す、ポインタが「ゼロ」であるかどうかを判断するため

の点検が行われる。

【0261】もしステップ1338において、点検評価されている現チャンネルが、合同ディレクトリの「最後の要素」を表さないと判断された場合、ステップ1340において、合同ディレクトリを形成するリンクされたリスト880内の次の要素に対応するチャンネルに合わせて「チャンネル」の値が望ましくは増値される。

【0262】加えて、ステップ1342において、合同ディレクトリからマウントを受けているチャンネル内のポイント(マウント合同ポイント)635の値が望ましくは、「チャンネル」によって指される合同ディレクトリ内の現ディレクトリを表示するように更新される。その後、プロセスはステップ1332に戻って上に述べた仕方

で処理を継続する。
【0263】しかし、もしステップ1338において、点検評価されている現チャンネルが、合同ディレクトリの「最後の要素」を表すと判断された場合には、バスネームアーギュメントによって表示されるファイルは、受信された「チャンネル」アーギュメントによって表されるチャンネルにはない。すなわち、そのチャンネルにマウントされた合同ディレクトリ内にある。したがって、「ウオーク」は不成功で、「ウオーク」機能1300は望ましくは、ステップ1345において「ゼロ」の値を持って呼び出し側の作業に戻る。

【0264】上に述べたように、「ウオーク」機能1300は、図16に示すデバイスウオークサブルーチンを利用して、受信されたチャンネルによって定まる或る特定のディレクトリが、受信されたバスネーム「要素」によって特定される或る表示されたファイルを有するかどうかを判断する。加えて、受信されたチャンネルが、「ウオーク」成功時に新たなファイルを指すので、デバイスウオークサブルーチン1400は又、要すれば、適切なバスデータ構造1140(図12(C))を更新する。

【0265】前に述べたように、「ウオーク」機能1300は、デバイステーブル450にアクセスして、ディレクトリを提供する核デバイスに適した特定のデバイスウオークサブルーチン1400を実行する。すなわち、各デバイスウオークサブルーチン1400は、その特定の核デバイスについての「ウオーク」機能を実現するのに必要な特定のコードを有する。デバイスに特定のコードは、ここでは関係ない。そのため、図16に示すデバイスウオークサブルーチン1400は、本発明に関連する概念を例示するように一般化されている。

【0266】図16に示すように、デバイスウオークサブルーチン1400は、ステップ1405においてプロセスに入って、処理を開始する。その後、受信された「チャンネル」アーギュメントによって表されるディレクトリが、受信されたバスネーム「要素」アーギュメントによって特定されるファイルを有するかどうかを判断するのに必要な、デバイスに特定のコードが、ステップ1

410において実行される。

【0267】ステップ1415において、そのバスネーム「要素」アーギュメントが、表示されたディレクトリ内に見出されたかどうかを判断するための点検が行われる。もしステップ1415において、そのバスネーム「要素」アーギュメントが、表示されたディレクトリ内に見出されなかったと判断された場合、デバイス「ウオーク」は不成功だった。そして、デバイスウオークサブルーチン1400は、ステップ14205において、望ましくは「ゼロ」の値を持って、呼び出し側の「ウオーク」機能1300に戻る。

【0268】しかし、もしステップ1415において、そのバスネーム「要素」アーギュメントが、表示されたディレクトリ内に見出されたと判断された場合、デバイス「ウオーク」は成功だった。そしてステップ1430~1460において、受信された「チャンネル」が今や受信されたバスネーム「要素」によって表示されるファイルを指すことを表示するように、適切なデータ構造を更新する必要がある。

【0269】すなわち、ステップ1425において、受信された「チャンネル」の要素615が、望ましくはそのバスネーム「要素」アーギュメントによって特定されたファイルの「qid」値を反映するように更新される。なお、「qid」値は、そのファイルを与えるデバイスによって出され、ステップ1410におけるデバイスに特定のコードの実行時に取得される。

【0270】今や、じゅしんされた「チャンネル」が新たなファイルを指しているので、そのファイルに関連するバスデータ1140を指すチャンネル内のポイントも同様に更新する必要がある。加えて、もし受信された「チャンネル」によって今指されているファイルが前にアクセスされていない場合、バスデータコード1140を形成する必要がある。

【0271】前に述べたように、サーバ樹木バスリスト1110(図12(B))は、そのファイルを与えるデバイスに関連するバスデータ構造の全てを指すポイントのリストを有する。サーバ樹木バスリスト1110は望ましくは、バスネーム「要素」と親のファイルのバスデータ構造を指すポイントとによってハッシュ処理される。そして、ステップ1430において、適切なハッシュ値が利用されてバスデータ構造の適切なリンクされたリスト1126がアクセスされる。

【0272】ステップ1435において、リンクされたリスト1126内のデータ構造が、受信されたバスネーム「要素」アーギュメントに等しいファイルネームを有するかどうか、及びバスデータ構造1140の要素1144から取り出された、親のバスデータ構造値を指すポイントが、受信された「チャンネル」アーギュメントの要素640(図7)内に表示されたバス構造値を指すポイントに等しいかどうか、を判断するための点検が行われ

る。

【0273】もしステップ1435において、リンクされたリスト1126内のデータ構造が、受信されたバスネーム「要素」アーギュメントに等しいファイルネームを有さず、且つ親のバスデータ構造値1144を指すポインタが、受信された「チャンネル」アーギュメントの要素640内のバス構造値を指すポインタに等しくない、と判断された場合、現ファイルは前にアクセスされておらず、又バスデータ構造はそのファイルについては前に形成されていない。

【0274】したがって、ステップ1440において、望ましくは、そのファイルについてバスデータ構造1140が形成され、ファイルネーム要素1142に、受信されたバスネーム「要素」アーギュメントが入れられ、親のバスデータ構造を指すポインタ1144が、受信された「チャンネル」アーギュメント等しくされる。ステップ1440において形成されたバスデータ構造1140は、ステップ1430においてアクセスされたバスデータ構造のリンクされたリスト1126に挿入される（ステップ1445）。

【0275】ステップ1445は例えば、リンクされたリストを指すポインタ1128を、バス樹木リスト1110の縦列1124内のエントリからステップ1440において形成されたデータ構造1140の最後の要素1148へコピーすることによって、そして新たに形成されたデータ構造を指すポインタを、ステップ1430においてアクセスされたサーバ樹木バスリスト1110の縦列1124内のエントリへコピーすることによって、実現される。その後、プロセスはステップ1450に進み、下で述べる仕方では処理を継続する。

【0276】しかし、もしステップ1435において、リンクされたリスト1126内のデータ構造が、受信されたバスネーム「要素」アーギュメントに等しいファイルネームを有し、且つ親のバスデータ構造値1144を指すポインタが、受信された「チャンネル」アーギュメントの要素640内のバス構造値を指すポインタに等しい、と判断された場合、バスデータ構造がそのファイルについて前に形成されている。

【0277】したがって、受信された「チャンネル」内のバス構造ポインタ640が望ましくは、ステップ1450において、受信されたバスネーム要素アーギュメントによって表示された、ウオークされた先（ウオーク動作の終端場所）のファイルのバスデータ構造を指すように更新される。

【0278】受信された「チャンネル」内のバス構造ポインタ640は今や、ウオークされた先のファイルに関連するバスデータ構造を指し、ウオークされた元（ウオーク動作の発端場所）のファイルに関連するバスデータ構造をもはや指さない。

【0279】それで、ウオークされた先のファイルに関

連するバスデータ構造1140内の基準カウンタ1146がステップ1455において増値され、ウオークされた元のファイルに関連するバスデータ構造1140内の基準カウンタ1146がステップ1460において減値される。この仕方では、各構造を現在指すポインタの数のカウントが維持される。

【0280】その後、プロセスは望ましくはステップ1475において、変数「チャンネル」の現在値を持って「ウオーク」機能に戻る。

10 【0281】前に述べたように、核400は、図17に示す「domount」（ドゥーマウント）機能1500を有し、このドゥーマウント機能1500は、「チャンネル」アーギュメントを受信し、マウントテーブル800にアクセスして、受信された「チャンネル」アーギュメントによって表されるチャンネルが「他からマウントを受けて」いるかどうかを判断する。

【0282】もしドゥーマウント機能1500が、変数「チャンネル」の現在値に対応するチャンネルが「他からマウントを受けて」いると判断した場合、変数「チャンネル」のその値が、マウントされたチャンネルを表示するように翻訳変換される。ドゥーマウント機能1500は、図17に示すように、ステップ1505においてプロセスに入り、受信された「チャンネル」をアーギュメント（探索かぎ）として処理を開始する。

【0283】ドゥーマウント機能1500は望ましくは、受信された「チャンネル」の要素615から「qid」値を取り出し、「qid」値に、適切な「ハッシュ」機能を実現して、ステップ1510において、マウント「from」データ構造の適切なリンクされたリスト826にアクセスする。ステップ1515において、マウント「from」データ構造によって指されるチャンネルが、受信された「チャンネル」アーギュメントと同じ「qid」値を有するかどうかを判断する点検が行われる。

【0284】もしステップ1515において、マウント「from」データ構造によって指されるチャンネルが、受信された「チャンネル」アーギュメントと同じ「qid」値を有すると判断された場合、マウント「to」ポインタ845に従って、これに対応するマウント「to」データ構造850に到達することにより、マウントされたチャンネルを指すポインタ855が取り出される。

【0285】それから、ステップ1520において、マウントされたチャンネルを指すこのマウント「to」ポインタ855によって指されるチャンネルがクローン処理（コピー）され、変数「チャンネル」に再割り当てされる。なお、マウントテーブル800内のチャンネルは望ましくは、クローン処理されたチャンネルがマウントテーブル800に影響を与えないように、タスク実行の前にクローン処理される。

50 【0286】しかし、もしステップ1515において、

マウント「from」データ構造830によって指されるチャンネルが、受信された「チャンネル」アーギュメントと同じ「qid」値を有しないと判断された場合、そのチャンネルは、「他からマウントを受けて」いない。そしてステップ1525において、チャンネル値が改変なしに戻される。

【0287】[遠隔のポインタへの処理タスクのエクスポート]前に述べたように、分散形計算環境100のような分散形システムにおいて、ユーザは、コンパレーションのような集中計算アプリケーションを、例えばcpuコマンドを実行することにより、cpuサーバ120のような遠隔のプロセッサへ移出(エクスポート)する。このcpuコマンドをユーザが実行後、コマンドインタプリタからのコマンドプロンプトが、ユーザ端末105のような端末の表示ウインドウに現れる。しかし、コマンドインタプリタはcpuサーバ上で処理を実行中である。

【0288】下で更に述べるように、cpuコマンドは、端末をして、ローカルのネーム空間の正確な記述をcpuサーバ120にエクスポートさせる。cpuサーバ120は受信したネーム空間を利用して、端末のネーム空間とはほぼ同一のネーム空間を組み立てる。この仕方

で、cpuサーバ120は、端末によって組み立てられたのと同じカスタム化した構成図をこの分散形計算環境100について見ることになる。

【0289】一推奨実施例において、cpuコマンドが、図18に示すプロセス実行エクスポート1600(プロセス実行作業を移出するための処理プログラム)の実行をユーザ端末105上に呼び出す。プロセス実行エクスポート1600がメッセージを、選択されたcpuサーバ120に送信すると、この選択されたcpuサーバ120が、図19に示すプロセス実行インポート1650の実行をこの選択されたcpuサーバ120上に呼び出す。

【0290】下で述べるように、プロセス実行エクスポート1600及びプロセス実行インポート1650の実行には、2個の遠隔のプロセッサ上の2個のプロセス間の相互対話処理が必要である。これに関連して、図18及び図19の流れ図において種々の計算システム間の信号の転送を示すために、次の取り決めを設定する。

【0291】すなわち、或るプロセッサが信号を別のプロセッサに送信しているとき、そのプロセスに対応するステップを図18のステップ1610のような右側が突出した矢印形ブロックで示す。同様に、或るプロセッサが別の端末から送信された信号を受信しているとき、そのプロセスに対応するステップを図19のステップ1660のような「フラッグ」形ブロックで示す。

【0292】図18に示すように、プロセス実行エクスポート1600は、ステップ1605において、例えばユーザまたはプロセスによるcpuコマンドの実行時に

プロセスに入って処理を開始する。その後、プロセス実行エクスポート1600は、ステップ1610において、cpuサーバ120のような選択された遠隔のプロセッサに既知の仕方

でダイヤルする。本発明の一態様によれば、ローカルのネーム空間の現在の状態の正確な表現が生成され、この選択されたサーバ120に送信される。

【0293】それから、ステップ1620において、図20に関連して下で述べるネーム空間読み出しサブルーチン1700が実行される。前に述べたように、ローカルのネーム空間は、一連の「連結」及び「マウント」コマンドによって形成される。「連結」及び「マウント」コマンドの各々は、マウントテーブル800内の個々のエントリによって実現される。

【0294】すなわち、ネーム空間読み出しサブルーチン1700が、マウントテーブル800にアクセスして、「連結」及び「マウント」コマンドの各々に連関する「from」ファイルと「to」ファイルとからなる「ネーム空間リスト」を生成する。又、これらのファイルによってローカルのネーム空間の現在の状態が形成される。

【0295】なお、ネーム空間読み出しサブルーチン1700の推奨実施例では、「ネーム空間リスト」の生成は、その特定のファイルを与えるサーバのネーム空間内のバスネームを用いて行われ、ローカルのネーム空間は必ずしも用いられない。

【0296】次に、ステップ1625において、図21に示すバスネーム置換サブルーチン1800が望ましくは実行されて、ネーム空間読み出しサブルーチン1700から返された「ネーム空間リスト」が分析され、この特定のファイルを与えるサーバのネーム空間内の「from」バスネーム及び「to」バスネームが、プロセスのローカルのネーム空間全体の、完全なバスネームに置換される。

【0297】なお、「ネーム空間リスト」内の「from」バスネーム及び「to」バスネームはユーザまたはプロセスのローカルのネーム空間に完全に指定されているが、バスネーム置換サブルーチン1800の実行後に、「ネーム空間リスト」内のエントリを更に分析して各エントリが「連結」コマンド又は「マウント」コマンドに連関するかどうかを判断する必要がある。

【0298】更に又、エントリが「マウント」コマンドの結果である場合、非核ファイルシステムのファイル樹木が、ローカルのネーム空間に既にあるファイルにマウントされている。したがって、図8(A)に関連して上で述べたように、非核ファイルシステムとの通信リンクをパイプ又はネットワーク接続を用いて設立する必要がある。

【0299】そこで、ステップ1630において、プロセス実行エクスポート1600が、図22、図23

(A)、及び図23(B)に示すネーム空間パッケージ化サブルーチン1900を実行する。

【0300】このネーム空間パッケージ化サブルーチン1900は、「ネーム空間リスト」内のエントリを分析して、ステップ1610において設立されたネットワーク接続を介して上記の選択されたcpuサーバ120に送信するために、これらのエントリを「連結」コマンド及び「マウント」コマンドのリストとして一体のパッケージにまとめる(パッケージ化する)処理を行う。

【0301】加えて、「マウント」コマンドに対応する「ネーム空間リスト」内の各エントリについてネーム空間パッケージ化サブルーチン1900が、マウントされた非核ファイルシステムへの適切な通信リンクのアクセス又は設立の仕方についての情報をcpuサーバ120に供給する。

【0302】その後、ネーム空間パッケージ化サブルーチン1900によって生成された「ネーム空間パッケージリスト」が、ステップ1635において、この選択されたcpuサーバ120に送信される。端末がローカルのネーム空間の表現をcpuサーバ120にエクスポートし終った後に、ステップ1640において、プロセス実行エクスポート1600はファイルサーバとなり、ネットワーク接続を介して受信されるcpuサーバ120からの遠隔手順コール要求に応答する。

【0303】本質的に、プロセス実行エクスポート1600が、この選択されたcpuサーバ120に対してファイルサーバとして機能しているとき、プロセス実行エクスポート1600は、cpuサーバ120のネーム空間にマウントされている図8(A)に示す非核ファイルシステム750のような、非核ファイルシステムである。

【0304】前に述べたように、ユーザが、集中計算アプリケーションのような処理タスクを、cpuサーバ120のような遠隔のプロセッサにエクスポートしたいと望む場合、ユーザ端末105上で実行中のプロセス実行エクスポート1600がステップ1610において、cpuサーバ120のような選択された遠隔のプロセッサにダイヤルする。

【0305】望ましくは、cpuサーバ120は、図19に示すプロセス実行インポート1650を有する。このプロセス実行インポート1650は、図19のステップ1660に示すように、既知の仕方、ネットワークを介して受信された通信を絶えずモニタして、到来する接続要求を聞き取るためのプロセスを有する。

【0306】ユーザ端末105上で実行中のプロセス実行エクスポート1600から接続要求を受信すると、cpuサーバ120が望ましくは、端末105上でコマンドを処理するために、ステップ1662において、この端末用の新たなプロセスグループを形成する。なお、プロセスグループは、同じネーム空間を共用する1個以上

のプロセスからなる。

【0307】その後、ステップ1665においてcpuサーバ120は、ステップ1635において端末によって送信された「ネーム空間パッケージリスト」を待つ。「ネーム空間パッケージリスト」を受信すると、プロセス実行インポート1650が、ステップ1670において、端末のルートディレクトリを表す、ファイル記述子「fd」によって特定されるチャンネルを形成する。それから、プロセス実行インポート1650は、ステップ1675において、新たなネーム空間を形成し、古いネーム空間を廃棄する。

【0308】ステップ1670において形成された、端末105のルートディレクトリを表すチャンネルが望ましくは、cpuサーバ120の新たなネーム空間で、端末に割り当てられたネーム空間の一部分を表す位置「/mnt/term」にマウントされる。

【0309】図5及び図8(A)に関連して上で述べたように、遠隔のファイルシステムのルートディレクトリを表すチャンネルのマウント処理は望ましくは、cpuサーバ120のマウントデバイス470をして、ユーザ端末への通信リンクを表すマウントポイントを設立させる。図8(A)に示すチャンネルデータ構造705が端末105のルートディレクトリを表す。一方、非核ファイルシステム750がユーザ端末105である。

【0310】その後、ステップ1685において、プロセス実行インポート1650が、ステップ1665において端末から受信された「ネーム空間パッケージリスト」内の「連結」および「マウント」コマンドのリストを実行する。cpuサーバ120は、受信したネーム空間を利用して、端末105のネーム空間にほぼ同一のネーム空間を組み立てる。この仕方、cpuサーバ120は、端末によって組み立てられたのと同じカスタム化した構成図をこの分散形計算環境100について見ることになる。

【0311】いい替えれば、プロセス実行インポート1650が端末105のネーム空間内にあったリソースを、ステップ1675において形成された、新たなcpuネーム空間内の同じ場所へ連結する。例えば、ユーザ端末105上のマウス装置を表すファイル「/mnt/term/dev/mouse」が、ファイル「/dev/mouse」に連結される。

【0312】そして、cpuによる、ファイル「/dev/mouse」にアクセスするための次の試みが、cpuサーバ120内のマウントデバイス470によって、端末105に送られる遠隔手順コールメッセージに翻訳変換される。こうして、マウス及び表示モニタのような端末105上のローカルのリソースが、cpuサーバ120上で端末105と同じネーム空間内の場所で実行中のプロセスに見えるようになる。

【0313】端末から受信されたネーム空間パッケージ

がcpuサーバによって処理され終ると、プロセスは、ステップ1690において、シェルプロセスを開始して、コマンドインタプリタを始める。プロセス実行インポート1650の実行は、ユーザによる「ファイルの終了」コマンドの実行に続いて最後のチャンネルを閉じるときにステップ1695において終結する。

【0314】前に述べたように、プロセス実行エクスポート1600は、ステップ1620において、ネーム空間読み出しサブルーチン1700を実行する。ネーム空間読み出しサブルーチン1700は、マウントテーブル800にアクセスして、ローカルのネーム空間の現在の状態を形成するために実行された「連結」及び「マウント」コマンドの各々に連関する「from」ファイルと「to」ファイルとからなるネーム空間リストを生成する。

【0315】図20に示すように、ネーム空間読み出しサブルーチン1700は、ステップ1705において、例えばユーザまたはプロセスによるcpuコマンドの実行時にプロセスに入って処理を開始する。ネーム空間読み出しサブルーチン1700は、ステップ1710において、変数「ネーム空間リスト」を初期化する。この変数を利用して、ネーム空間の現在の状態を表す、「連結」及び「マウント」コマンドの各々に連関するバスネームのリストが格納される。

【0316】その後、ネーム空間読み出しサブルーチン1700は、ステップ1720において、要素860に表示される最も低い「mountid」値を有するデータ構造850を求めてマウントテーブル800内のマウント「to」データ構造850の各々を探索する。前に述べたように、「mountid」値が、マウントテーブル800に新たなエントリが付加される都度、増値され、マウントテーブル800のエントリがその形成された順に取り出せるようにするタイムスタンプとして機能する。

【0317】それから、現在の「mountid」値に対応するマウントテーブル800内のエントリが、ステップ1730において、アクセスされる。ステップ1740において望ましくは、マウントテーブル800内の現エントリについての「ネーム空間リスト内にエントリを形成し、対応するマウント「from」データ構造から取り出された、「他からマウントを受けた」チャンネルを指すポイントと、マウント「to」データ構造から取り出された、マウントされたチャンネル及びマウント順序フラッグとを入れる。

【0318】その後、ステップ1760において、「他からマウントを受けた」チャンネル及びマウントされたチャンネルについて、「チャンネル/バスネーム」コンバータ2000が個別に実行され、「ネーム空間リスト」内の形成されたエントリをこれらのファイルを提供するサーバのネーム空間内のチャンネルに対応するバスネームに置

換される。

【0319】ステップ1770において、マウントテーブル800内に点検評価すべきエントリがまだあるかどうかを判断する点検が行われる。もしステップ1770において、マウントテーブル800内に点検評価すべきエントリがまだあると判断された場合、ステップ1780において、次に最も低い「mountid」値について探索が行われる。その後、プロセスはステップ1730に戻って、上に述べた仕方では処理を継続する。

【0320】しかし、もしステップ1770において、マウントテーブル800内に点検評価すべきエントリがもはやないと判断された場合には、プロセスはステップ1790において、呼び出し側のプロセス実行エクスポート1600に戻る。

【0321】前に述べたように、プロセス実行エクスポート1600は、ステップ1625において、バスネーム置換サブルーチン1800を実行して、ネーム空間読み出しサブルーチン1700によって形成された「ネーム空間リスト」を分析し、それぞれのサーバのネーム空間内の「from」バスネーム及び「to」バスネームをユーザまたはプロセスのローカルのネーム空間全体の完全なバスネームに置換する。なお、「ネーム空間リスト」内の各エントリは、連関する「連結」又は「マウント」コマンドに対応する1対の「from」バスネーム及び「to」バスネームの要素からなる。

【0322】図21に示すように、バスネーム置換サブルーチン1800はステップ1805でプロセスに入る。その後、ステップ1810において、バスネーム要素が「ネーム空間リスト」から取り出される。「ネーム空間リスト」内のエントリで現バスネーム要素より先行するものがもしあれば、その要素がステップ1815において取り出される。

【0323】ステップ1820において、非核ファイルシステムのネーム空間がマウントされたマウントポイントを一般に指定する現バスネーム要素のルートが、現前回（現時点で（すぐ）前）のエントリ内の「from」バスネーム要素に等しいかどうかを判断するための点検が行われる。

【0324】もしステップ1820において、現バスネーム要素のルートが、現前回のエントリ内の「from」バスネーム要素に等しいと判断された場合、ステップ1825において、現バスネーム要素のルートが、現前回のエントリ内の、対応する「to」バスに置換される。

【0325】しかし、もしステップ1820において、現バスネーム要素のルートが、現前回のエントリ内の「from」バスネーム要素に等しくないと判断された場合には、ステップ1830において、「ネーム空間リスト」内に、現バスネーム要素に対して点検評価すべき前のエントリがまだあるかどうかを判断するための点検

が行われる。

【0326】もしステップ1830において、「ネーム空間リスト」内に、現バスネーム要素に対して点検評価すべき前のエントリがまだあると判断された場合、ステップ1835において、現前回のエントリよりも次に前の（より前の）エントリが「ネーム空間リスト」から取り出される。その後、プロセスはステップ1820に戻って、この、次に前のエントリを、上に述べた仕方

で処理する。
【0327】しかし、もしステップ1830において、「ネーム空間リスト」内に、現バスネーム要素に対して点検評価すべき前のエントリがもはやないと判断された場合には、ステップ1840において、「ネーム空間リスト」内に点検評価すべきバスネーム要素がまだあるかどうかを判断するための点検が行われる。

【0328】もしステップ1840において、「ネーム空間リスト」内に点検評価すべきバスネーム要素がまだあると判断された場合、ステップ1845において、次の要素が「ネーム空間リスト」から取り出される。その後、プロセスはステップ1815に戻って、この、次のバスネーム要素を、上に述べた仕方

で処理する。
【0329】しかし、もしステップ1840において、「ネーム空間リスト」内に点検評価すべきバスネーム要素がもはやないと判断された場合には、プロセスはステップ1850において、修正された「ネーム空間リスト」を呼び出し側のプロセス実行エクスポート1600に返す。

【0330】前に述べたように、プロセス実行エクスポート1600は、ステップ1630（図18）において、ネーム空間パッケージ化サブルーチン1900を実行して、「ネーム空間リスト」内のエントリを取り出し、遠隔のcpuサーバに送信するために、これらのエントリを「連結」コマンド及び「マウント」コマンドのリストとしてパッケージ化する処理を行う。

【0331】図22に示すように、ネーム空間パッケージ化サブルーチン1900は、ステップ1904においてプロセスに入る。ネーム空間パッケージ化サブルーチン1900は、ステップ1908において、変数「ネーム空間パッケージリスト」を初期化する。この「ネーム空間パッケージリスト」は、cpuサーバ120に送信されることになる「連結」及び「マウント」コマンドのリストを格納する。

【0332】ステップ1912において、望ましくは、「ネーム空間リスト」内のエントリが取り出される。ステップ1916において、「他からマウントを受けた」ファイルに対応する「from」バスネーム要素が点検されて、「ネーム空間リスト内の現エントリの、「連結」コマンド又は「マウント」コマンドとの関連の有無が判断される。なお、「連結」コマンドは、この「from」バスネームがマウントされたパイプ又はネットワ

ーク接続ではないことを検証することによって特定される。

【0333】なお又、マウントされたパイプ又はネットワーク接続に関連する「from」バスネームは、パイプ又はネットワーク接続を可能にするデバイス有する。もしステップ1916において、「ネーム空間リスト」の現エントリが「連結」コマンドに対応すると判断された場合、ステップ1920において「ネーム空間リスト」の現エントリから取り出された「from」及び「to」バスネームからなる「連結」コマンドが「ネーム空間パッケージリスト」に付加される。その後、プロセスは下で述べるステップ1990に進む。

【0334】しかし、もしステップ1916において、「ネーム空間リスト」の現エントリが「連結」コマンドに対応しないと判断された場合、このエントリは「マウント」コマンドに対応するに違いない。そして、「from」バスネームは非核ファイルシステムへのパイプ又はネットワーク接続を表す。それで、ステップ1924において、この「from」バスネームが、マウントされたパイプに現エントリが対応すると表示しているかどうかを判断する点検が行われる。

【0335】なお、もしこの「from」バスネームが、マウントされたパイプを表示している場合、このバスネームは一般に、パイプを与える非核デバイスを有する。更に又、パイプは、下で述べるような、「/srv」ディレクトリに表示される、名前付きパイプの場合、又名前付きでないパイプの場合が有り得る。

【0336】そして、もしステップ1924において、この「from」バスネームが、マウントされたパイプに現エントリが対応すると表示していると判断された場合、ステップ1928において、「qid」及び「種類」の値が、この「他からマウントされた」「from」バスネームに対応するチャンネルから取り出される。

【0337】その後、ステップ1932において、「/srv」ディレクトリ内のファイルが、同じ「qid」及び「種類」の値を有するチャンネルを指すかどうかを判断するための点検が行われる。

【0338】もしステップ1932において、「/srv」ディレクトリ内のファイルが、同じ「qid」及び「種類」の値を有するチャンネルを指すと判断された場合、パイプはネーム付きのパイプである。そして、ステップ1936において、「マウント」コマンドが「ネーム空間パッケージリスト」に付加される。この「マウント」コマンドは、「to」バスネームによって表示されるファイルに、ファイル「/mnt/term/srv/filename」が、マウントされるべきことを表示するコマンドである。

【0339】又、「filename」は、ファイルが、ステップ1932において特定された「/srv」ディレクトリ内のファイルを示す。この仕方

サーバ120が、表示された「マウント」コマンドを実行するとき、ユーザ端末105上の同じパイプを利用して、このパイプを介してマウントされた非核ファイルシステムがアクセスされる。

【0340】しかし、もしステップ1932において、「/srv」ディレクトリ内のファイルが、同じ「qid」及び「種類」の値を有するチャンネルを指さないと判断された場合には、このパイプは名前付きではない。したがって、パイプを表すチャンネルを指すポインタを格納するために、ステップ1940（図23（A））において、ファイルが、或る特定のファイルネームを付けて形成される。この方法により、パイプを有効に名前付きパイプにすることができる。

【0341】なお、推奨する「ブラン9」形オペレーティングしにおいては、ファイル#Mnについて「開設」コマンドを実行することにより（#Mはマウントデバイス470（図5）を示し、nは現「qid」値に等しい）、結果としてチャンネル自体がパイプを表す。これは、「/rsrv」ディレクトリ内の形成されたファイルに掲示される。

【0342】その後、望ましくはステップ1948において、「マウント」コマンドが「ネーム空間パッケージリスト」に付加される。この「マウント」コマンドは、「to」バスネームによって表示されるファイルに、ファイル「/mnt/term/srv/filename」が、マウントされるべきことを表示するコマンドである。ここに、「filename」は、ステップ1940において形成されたファイルを示す。

【0343】この仕方では、cpuサーバ120が、表示された「マウント」コマンドを実行するとき、ユーザ端末105上の同じパイプを利用して、このパイプを介してマウントされた非核ファイルシステムがアクセスされる。

【0344】しかし、ここで注記したいのは、ステップ1940においてファイルを形成しようとする試みが核400によっていつも許されるとは限らないことである。例えばアクセス許可が変更される場合がある。そのため、望ましくはステップ1952において、ステップ1940におけるファイル形成の試みが核エラーに遭遇しそのためファイル形成が不成功だったかどうかを判断するための点検が行われる。

【0345】もしステップ1952において、ステップ1940におけるファイル形成の試みが核エラーに遭遇しなかったと判断された場合、ファイル形成は成功で、プロセスは下で述べるステップ1990（図22）に進む。

【0346】しかし、もしステップ1952において、ステップ1940におけるファイル形成の試みが核エラーに遭遇したと判断された場合には、ファイル形成は不成功だった。したがって、望ましくはステップ1956

において、「/mnt/term」が、「ネーム空間リスト」の現エントリ内の「from」バスネームに付随される。

【0347】その後、望ましくは「連結」コマンドがステップ1960において「ネーム空間パッケージリスト」に付加される。この「連結」コマンドは、付随の「from」バスネームが、「to」バスネームによって表示されるファイルに連結されるべきことを表示するコマンドである。

10 【0348】加えて、図19に関連して下で述べるようにプロセス実行インポート1650がステップ1680において、cpuサーバ120のネーム空間内の「/mnt/term」に、端末のルートディレクトリを表すチャンネルをマウントする。

【0349】そして、cpuサーバ120が、付随しない「from」バスネームによって表示されるファイルへのアクセスを試みる場合、これが、付随する「from」バスネームによって表されるファイルへのアクセスに翻訳変換され、「/mnt/term」に連関するマ
20 ウントポイントを介してユーザ端末105へ行くことになる。その後、プロセスは下で述べるステップ1990（図22）に進む。

【0350】前に述べたように、ステップ1924（図22）において、「from」バスネームが、マウントされたパイプに現エントリが対応することを表示しているかどうかを判断するための点検が行われる。もしステップ1924において、「from」バスネームが、マウントされたパイプに対応せず、ネットワーク接続に対応すると判断された場合、プロセスはステップ1964
30 （図23（B））に進む。

【0351】なお、もし「from」バスネームがネットワーク接続を表す場合には、それは、図10（B）に関連して上で述べたように、「/net/il/2/data」の形式となろう。更に又注記したいのは、同じディレクトリ内の遠隔のファイルがコマンドのネットワーク接続に対応するネットワーク宛先アドレスを格納していることである。

【0352】そして、ステップ1964において、「from」バスネームが、「データ」ファイルから「遠隔」ファイルへ翻訳変換される。なお、「データ」ファイルバスネームが与えられると、「遠隔」ファイルへの変更は単に辞書的である。いい替えれば、「/net/il/2/data」のようなバスネームで始め、最後の斜線文字「/」において語「データ」の「遠隔」への置換を開始する。

【0353】その後、ステップ1968において、「遠隔」ファイルの内容が取り出される。「遠隔」ファイルから取り出されたネットワーク宛先アドレスが、それからステップ1972において、図10（A）に関連して
50 上で述べたように、ネットワークデータベース900を

用いて、対応する「マシンネーム及びサービス」に変換される。

【0354】その後、ステップ1976において、「マウント」コマンドが「ネーム空間パッケージリスト」に付加される。この「マウント」コマンドは、表示された「マシンネーム及びサービス」(machine!service)を「to」バスネームによって指定されたファイルにマウントするように、cpuサーバ120に命令するコマンドである。その後、プロセスは、下で述べるステップ1990(図22)に進む。

【0355】「ネーム空間リスト」からの現エントリが、いま述べた仕方で「ネーム空間パッケージリスト」にパッケージ化されると、ステップ1990において、「ネーム空間リスト」内にパッケージ化すべきエントリがまだあるかどうかを判断するための点検が行われる。もしステップ1990において、「ネーム空間リスト」内にパッケージ化すべきエントリがまだあると判断された場合、プロセスはステップ1912に戻って、上に述べた仕方で残りのエントリを処理する。

【0356】しかし、もしステップ1990において、「ネーム空間リスト」内にパッケージ化すべきエントリがもはやないと判断された場合、プロセスはステップ1995において、呼び出し側のプロセス実行エクスポート1600にもどる。

【0357】前に述べたように、図12に関連して上で述べた、各チャネルに関連するバスデータ構造1140が、与えられたチャネルのバスネームの生成を可能にするバス情報を供給する。

【0358】上で述べたように、ネーム空間読み出しサブルーチン1700がステップ1760(図20)において、「チャネル/バスネーム」コンバータ2000を実行して、与えられたチャネルを、対応するファイルを提供するサーバのネーム空間内のチャネルに対応するバスネームに置換する。図24に示すように、「チャネル/バスネーム」コンバータ2000は、ステップ2005において、分析すべきチャネルを指すポインタを受信してプロセスに入り、処理を開始する。

【0359】その後、ステップ2010において、「チャネル/バスネーム」コンバータ2000が、この受信されたポインタによって特定されたチャネルにアクセスする。そして、ステップ2015において、バス構造1140を指すポインタ640(図7)が、チャネル600のようなチャネルデータ構造から取り出される。このポインタ640に従って行くと、関連するバス構造1140に到達する。

【0360】ステップ2020において、変数「バスネーム」が最初に、バスデータの要素1142から取り出されたファイルネームにセットされる。その後、ステップ2025において、親のファイルに対応するバスデータ構造を指すポインタ1144(図12(C))が取り

出される。ポインタ1144に従うことによって、表示されたバスデータ構造1140に到達する。

【0361】ステップ2025において取り出された親のバス構造を指すポインタ1144が、サーバ樹木バステーブル1100(図12(A))の要素に表示されるルートディレクトリに関連するバスデータ構造を指すポインタに等しいかどうか、いい替えれば、ルートディレクトリに到達したかどうか、を判断するための点検がステップ2030において行われる。

10 【0362】もしステップ2030において、ステップ2025において取り出された親のバス構造を指すポインタが、ルートディレクトリに関連するバスデータ構造を指すポインタに等しくないとして判断された場合、現バスネームには取得すべき要素がまだあることになる。

【0363】したがって、ステップ2025においてアクセスされたバス構造に表示されるファイルネームが、ステップ2035において、変数「バスネーム」の先頭に、斜線文字で分け付けて付随される。その後、プロセスはステップ2025に戻って、バスネーム内の残りの要素を上で述べた仕方で処理する。

20 【0364】もしステップ2030において、ステップ2025において取り出された親のバス構造を指すポインタが、ルートディレクトリに関連するバスデータ構造を指すポインタに等しいと判断された場合には、それぞれのファイルサーバのルートディレクトリまでバスの全行程をさかのぼってルートディレクトリに到達したことになる。そしてステップ2040において、斜線文字「/」が変数「バスネーム」の先頭に付随される。要素が全て処理されたので、プロセスはステップ2050において呼び出し側のネーム空間読み出しサブルーチン1700に戻る。

【0365】以上の説明は、本発明の実施例及びその改変例に関するもので本発明の原理を例示したものに過ぎず、この技術分野の当業者であれば、本発明の種々の変形例を考え得るが、それらはいずれも本発明の技術的範囲に包含される。

【0366】

【発明の効果】以上述べたごとく、本発明によれば、複数のコンピュータ及びネットワークにかけて分散された分散形計算環境において、或る場所に所在する第1のプロセッサが、中央処理装置のような遠隔のプロセッサを呼び出して処理タスクを実行させることができる。

【0367】その際に、本発明によれば、従来技術で難点のあった、個々のユーザ又はプロセスによってカスタマイズされたネーム空間、に対する対応が容易となるので、遠隔のプロセッサが、第1のプロセッサから受信したネーム空間表現に基づいて修正されたネーム空間上でこの処理タスクを実行することが可能となる。したがって、分散形計算環境におけるタスク処理の効率を増大

【図面の簡単な説明】

【図1】本発明の機能を実現するのに適した分散形計算環境を示す説明図である。

【図2】図1の分散形計算環境においてユーザ端末として作動する汎用計算システムに関する説明図である。図2(A)に、同汎用計算システムの等角図を示し、図2(B)に、図2(A)の汎用計算システムに関連して利用されるマイクロプロセッサ処理システムのブロック図を示す。

【図3】ネーム空間を示す説明図である。図3(A)に、1個のユーザ端末と2個のファイルサーバとの個別ネーム空間を示し、図3(B)に、これら2個のファイルサーバのファイル樹木からなる、図3(A)のユーザのカスタム化されたネーム空間を示す。

【図4】本発明の機能を組み込んだオペレーティングシステムに関する説明図である。図4(A)に、本発明の機能を組み込んだオペレーティングシステムの核を示し、図4(B)に、デバイスに特定の手順コールを指すポインタのアレイを格納するために図4(A)の核が用いのに適したデバイステーブルを示す。

【図5】図1の分散計算環境における、核によって実現されないデバイス(非核デバイス)へのインタフェースとして図4(A)の核が用いのに適した核マウントデバイスを示す説明図である。

【図6】特定のプロセスについてのファイルアクセス情報を格納するのに適した、プロセスデータ構造及びファイル記述子アレイの部分図である。

【図7】特定のファイルを表現するため及びそのファイルに関連する情報を格納するために図4(A)の核によって利用される、チャンネルデータ構造を示す説明図である。

【図8】データ構造に関連する説明図である。図8(A)に、特定の非核ファイルシステムに対する図5のマウントポイントを共に形成する複数の関連データ構造を示す。図8(B)に、ユーザまたはプロセスのネーム空間を形成する「bind」(バインド)(連結)コマンド及び「mount」(マウント)(搭載)コマンドのシーケンスを格納するのに適した、マウントテーブル及び連関するデータ構造を示す。

【図9】図8(B)のマウントテーブル内のエントリを共に形成するマウント「from」データ構造と少なくとも1個のマウント「to」データ構造とを示す説明図である。

【図10】ネットワークデータベースに関連する説明図である。図10(A)に、図1の分散形計算環境における複数のマシンについてのネットワーク宛先アドレス情報を格納するのに適した、ネットワークデータベースを示す。図10(B)に、ネットワークに関連するユーザまたはプロセスのネーム空間の一部分を示す。

【図11】図5に示す非核ファイルシステムへの各通信

リンクについてのファイルを格納するのに適した、ユーザまたはプロセスのネーム空間内のディレクトリを示す説明図である。

【図12】サーバ樹木バスに関連する説明図である。図12(A)に、図5に示す非核ファイルシステムの樹木内のファイルの各々についてのバス情報を格納するのに適した、サーバ樹木バステーブルを示す。図12(B)に、図5に示す非核ファイルシステムの樹木内のファイルの各々についてのバスデータ構造を指すポインタを格納するのに適した、サーバ樹木バスリストを示す。図12(C)に、特定のファイルについてのバス情報を格納するのに適した、バスデータ構造を示す。

【図13】与えられたバスネームを、チャンネルとして表現される、運用すべき適切なファイル類似の対象物に翻訳するのに利用されるバスネームからチャンネルへの変換用のソフトウェアである「バスネーム/チャンネル」コンバータについて説明する流れ図の一部である。図14と共に本流れ図の完体を構成する。

【図14】与えられたバスネームを、チャンネルとして表現される、運用すべき適切なファイル類似の対象物に翻訳するのに利用される「バスネーム/チャンネル」コンバータについて説明する流れ図の一部である。図13と共に本流れ図の完体を構成する。

【図15】ファイル樹木の1つのレベルから別のレベルへ移動するために利用される「walk」(ウォーク)機能について説明する流れ図である。

【図16】特定のファイルについての与えられたデバイスのディレクトリをサーチするために図15の「ウォーク」機能によって利用されるデバイスウォーク・サブルーチンについて説明する流れ図である。

【図17】或る特定のチャンネルが「他からマウントを受けて」いるかどうかを判断するために、そしてもしそうなら、必要なチャンネル置換を行うために、図8Bのマウントテーブルにアクセスするのに利用される「dmount」(ドゥーマウント)機能について説明する流れ図である。

【図18】処理タスクをエクスポートする際に、処理タスクを、エクスポートする側のユーザ又はプロセスのネーム空間の正確な表現と共に遠隔のプロセッサへエクスポートするのに利用されるプロセス実行エクスポートについて説明する流れ図である。

【図19】図18においてエクスポートされる処理タスクを移入(インポート)するため及びエクスポートする側のユーザまたはプロセスのネーム空間を再生成するために図18のプロセス実行エクスポートに協力するプロセス実行インポートについて説明する流れ図である。

【図20】エクスポートする側のユーザまたはプロセスのネーム空間を形成する「連結」コマンド及び「マウント」コマンドのリストを生成するために図18のプロセス実行エクスポートが図8(B)のマウントテーブルに

61

アクセスするのに利用されるネーム空間読み出しサブルーチンについて説明する流れ図である。

【図21】図20のネーム空間読み出しサブルーチンによって生成された「連結」コマンド及び「マウント」コマンドのリストを分析するため、及びファイルを提供するサーバのネーム空間内のバスネームを図18のプロセス実行エクスポートのネーム空間全体の中の完全なバスネームに置換するために図18のプロセス実行エクスポートによって利用される、バスネーム置換サブルーチンについて説明する流れ図である。

【図22】遠隔のプロセッサへ送信するために図21のバスネーム置換サブルーチンによって修正変更される「連結」コマンド及び「マウント」コマンドのリストをパッケージ化するために図18のプロセス実行エクスポートによって利用される、ネーム空間パッケージサブルーチンについて説明する流れ図の一部である。図23(A)及び図23(B)と共に本流れ図の完体を構成する。

【図23】遠隔のプロセッサへ送信するために図21のバスネーム置換サブルーチンによって修正変更される「連結」コマンド及び「マウント」コマンドのリストをパッケージ化するために図18のプロセス実行エクスポートによって利用される、ネーム空間パッケージサブルーチンについて説明する流れ図の一部で、図23(A)及び図23(B)からなる。図22と共に本流れ図の完体を構成する。

【図24】ファイルを表すチャンネルをそのファイルの、対応するバスネームに置換するために図20のネーム空間読み出しサブルーチンによって利用される「チャンネル／バスネーム」コンバータについて説明する流れ図である。

【符号の説明】

100 分散形計算環境
105～110、305 ユーザ端末
120～121 中央処理装置(CPU)サーバ
130、131、310、315 ファイルサーバ
150 通信ネットワーク
200 汎用計算システム
202 外部ディスクドライブ
203 ハードディスクドライブ
204 表示モニタ
205 キーボード
206 処理装置
207 メモリ装置
208 マウス
209 ネットワークインタフェース
224、270 バス
250 制御装置
255 算術論理装置
260 ローカルメモリ装置

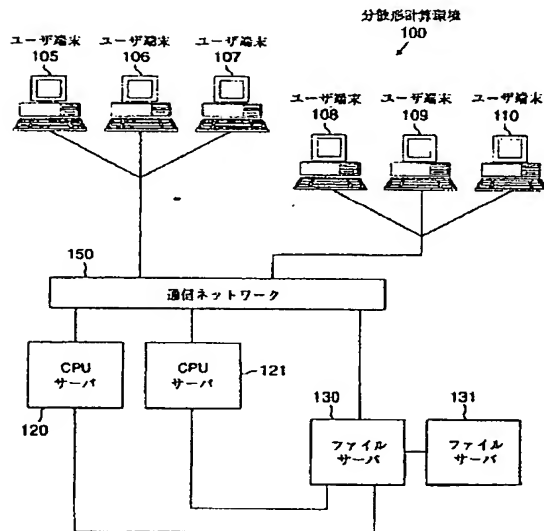
62

275 オペレーティングシステムプログラムコード
320 最小デフォルトネーム空間
325、330 ネーム空間
340 ネットワーク
400 核
410 プロセスマネージャ
420 ネーム空間マネージャ
450 デバイステーブル
470 核マウントデバイス
10 472～474 マウントポイント
476～478 通信リンク
480～482 非核ファイルシステム
500 ファイル記述子アレイ
502、504 縦列
520～522 チャンネル(チャンネルデータ構造)
550 プロセスデータ構造
555、560、565、605、610、615、620、625、630、635、640 要素(エントリ)
20 600 チャンネルデータ構造(チャンネル)
645 基準カウンタ
705、730 チャンネルデータ構造(又はチャンネル)
710 マウントポイントデータ構造(又は、マウントポイント)
712 チャンネルデータ構造ポインタ(又はポインタ)
714 待ち行列
716 ポインタ
740 通信リンク
750 非核ファイルシステム
30 800 マウントテーブル
805、810 縦列
815、820、825 横列
826、880 リンクされたデータ構造リスト
828、829 ポインタ
830 マウント「from」リンクリストデータ構造
832、837 マウント「from」データ構造
840、845、848 エントリ
850 マウント「to」リンクリストデータ構造
855、860、865、870、875 要素
40 900 ネットワークデータベース
950 ネーム空間
1000 ディレクトリ
1100 サーバ樹木バステーブル
1102 ポインタ
1126 リンクされたリスト
1140 バスデータ構造
1142、1144 エントリ
1146 基準カウンタ
1148 ポインタ
50 1200 「バスネーム／チャンネル」コンバータ

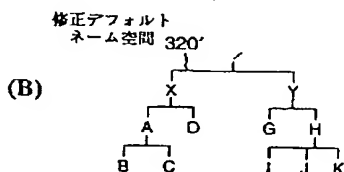
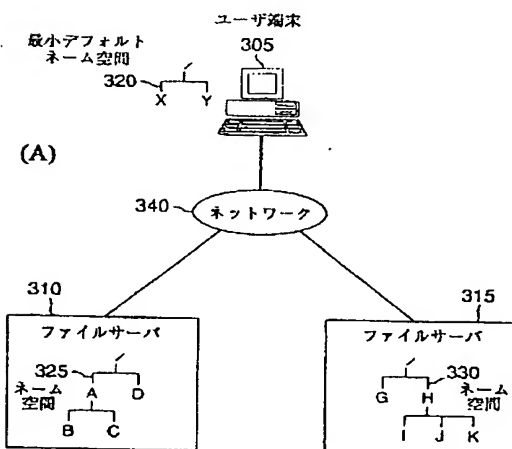
63

- 1245 「バスネーム／チャンネル」コンバータの第3部分
 1300 「walk」(ウォーク)機能
 1400 デバイスウォークサブルーチン
 1500 「domount」(ドゥマウント)機能
 1600 プロセス実行エクスポート *

【図1】



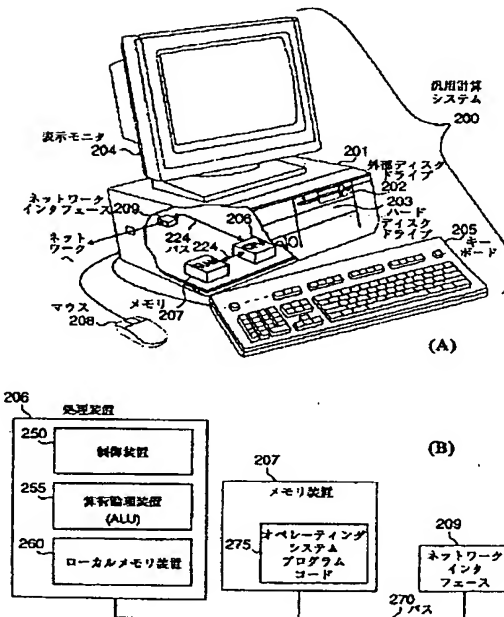
【図3】



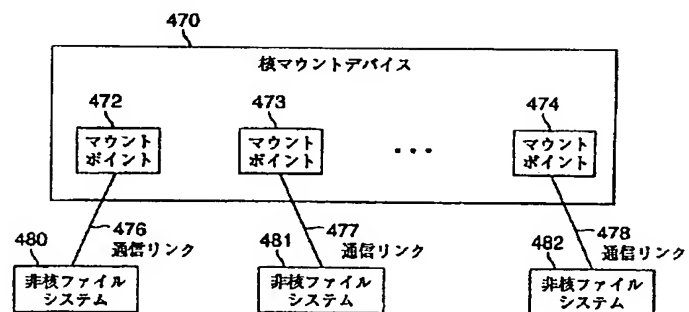
64

- * 1650 プロセス実行インポート
 1700 ネーム空間読み出しサブルーチン
 1800 バスネーム置換サブルーチン
 1900 ネーム空間パッケージ化サブルーチン
 2000 「チャンネル／バスネーム」コンバータ

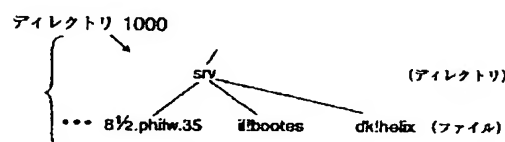
【図2】



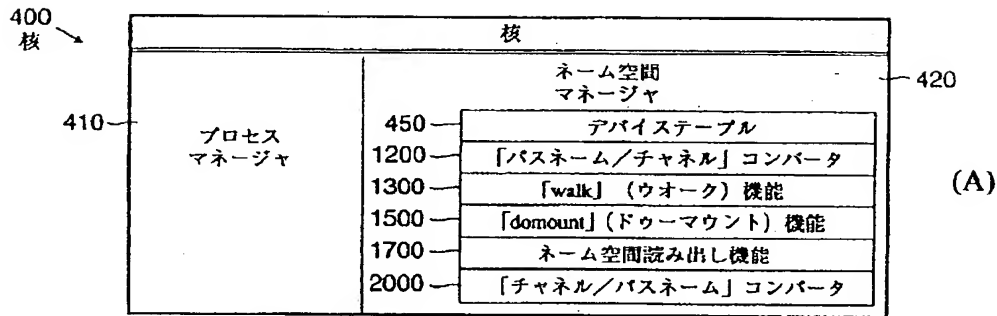
【図5】



【図11】



【図 4】



デバイステーブル

450

(B)

横列 461 →

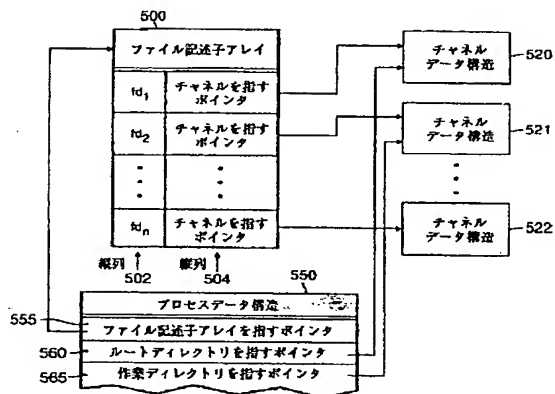
横列 462 →

横列 463 →

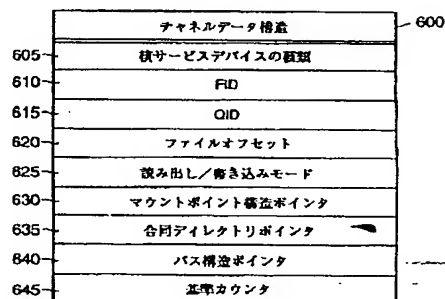
横列 464 →

デバイステーブル					
手順 デバイス	開く open	閉じる close	読み出す read	書き込む write	...
disk ディスク	diskopen (468)	diskclose	diskread	diskwrite	...
tty テレックス	ttyopen	ttyclose	ttyread	ttywrite	...
...
mount マウント	mntopen	mntclose	mntread	mntwrite	...
	452 縦列	453 縦列	454 縦列	455 縦列	456 縦列

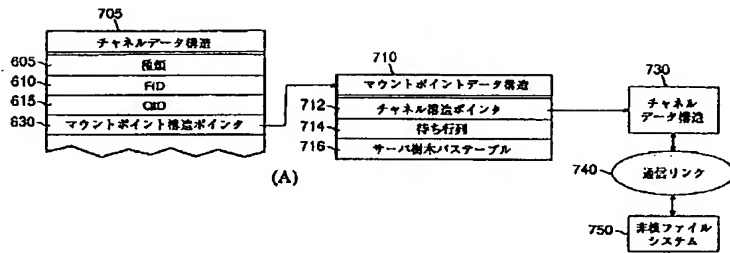
【図 6】



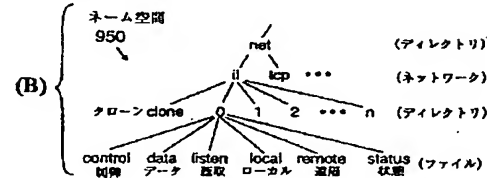
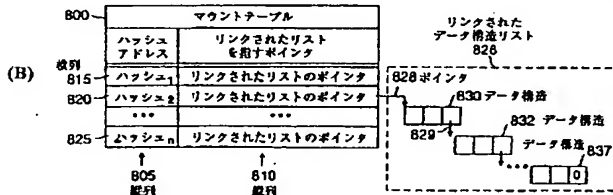
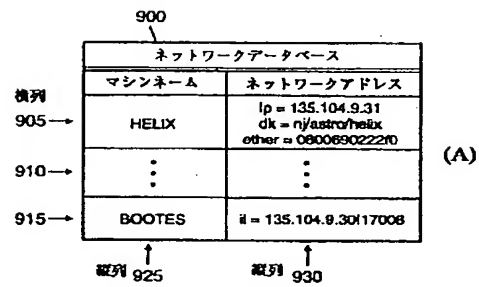
【図 7】



【図8】

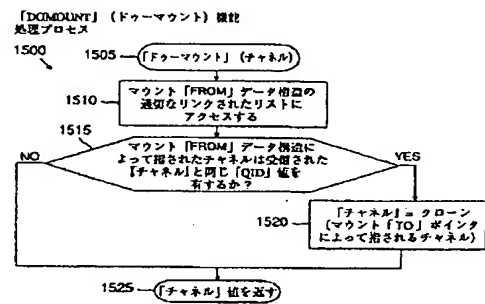
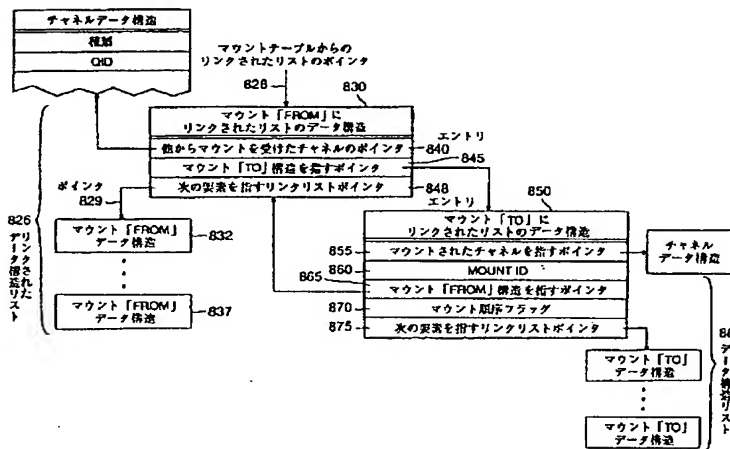


【図10】

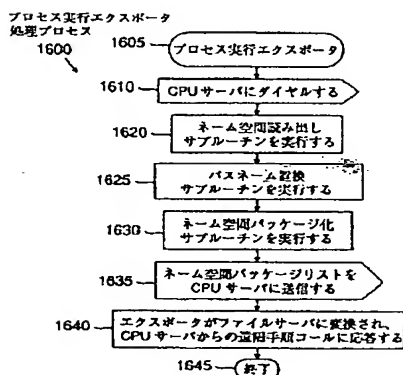


【図9】

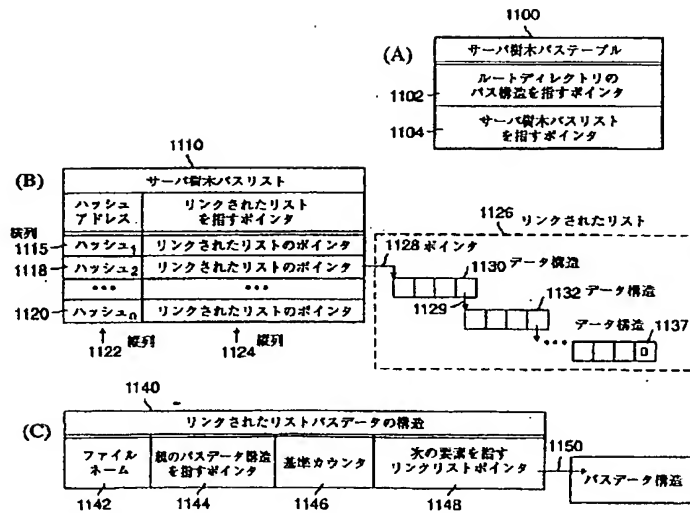
【図17】



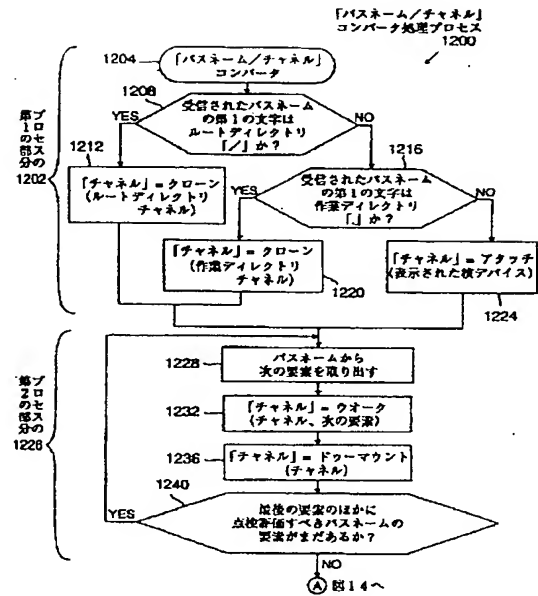
【図18】



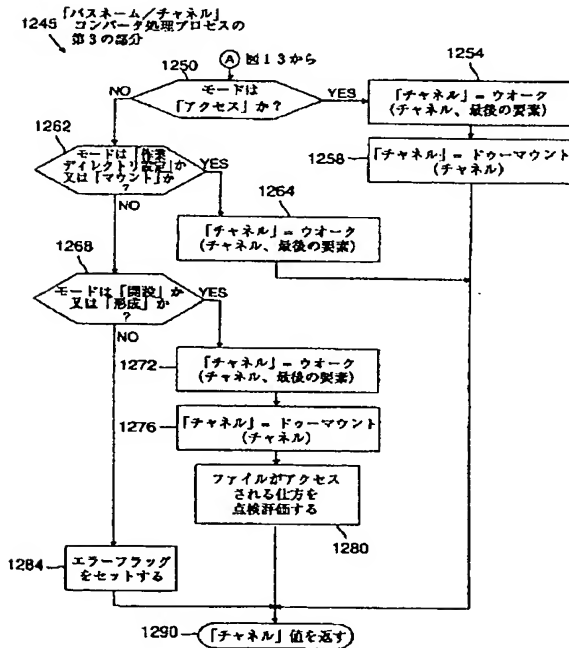
【図12】



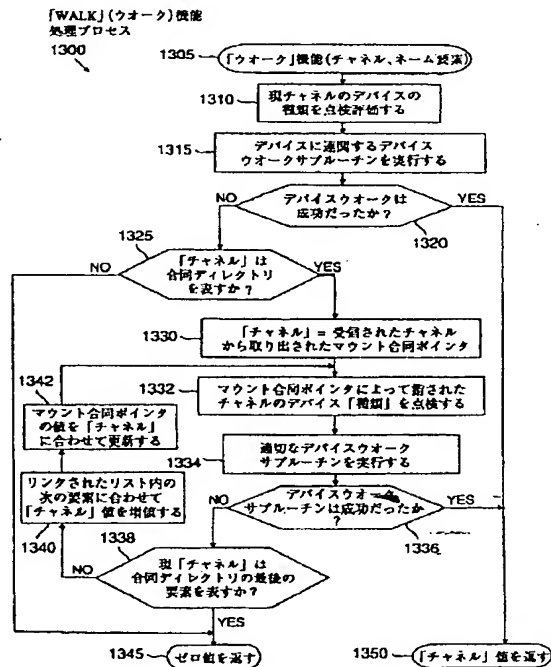
【図13】



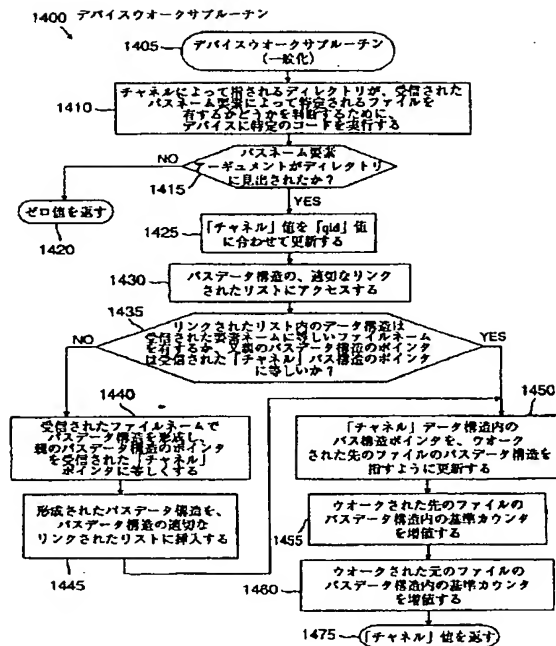
【図14】



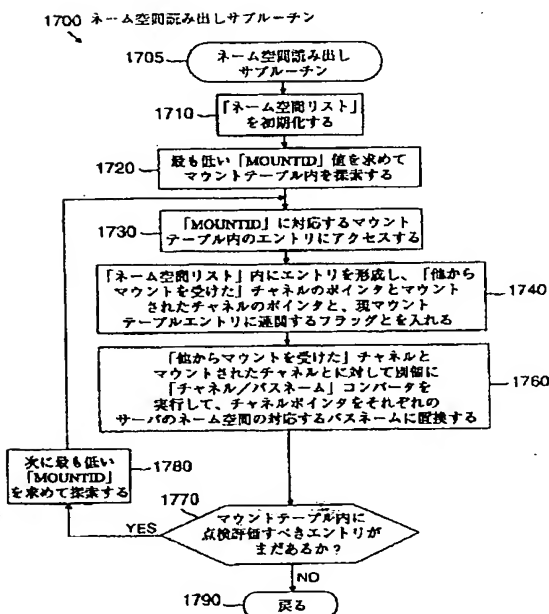
【図15】



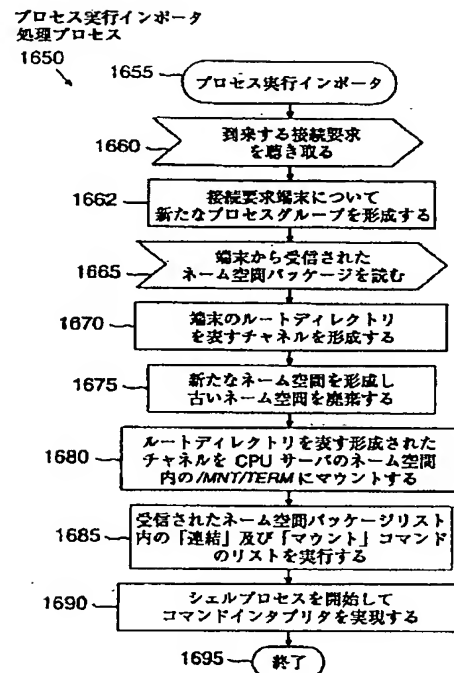
【図16】



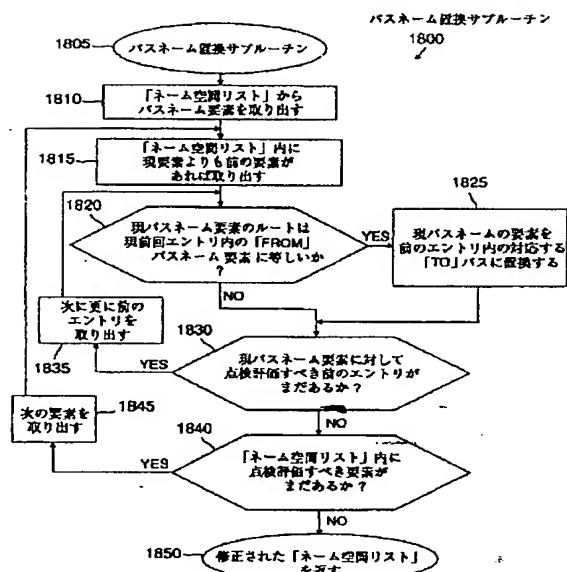
【図20】



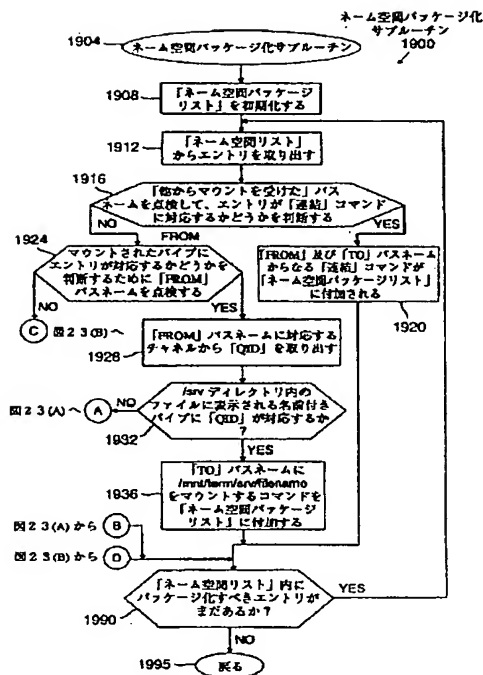
【図19】



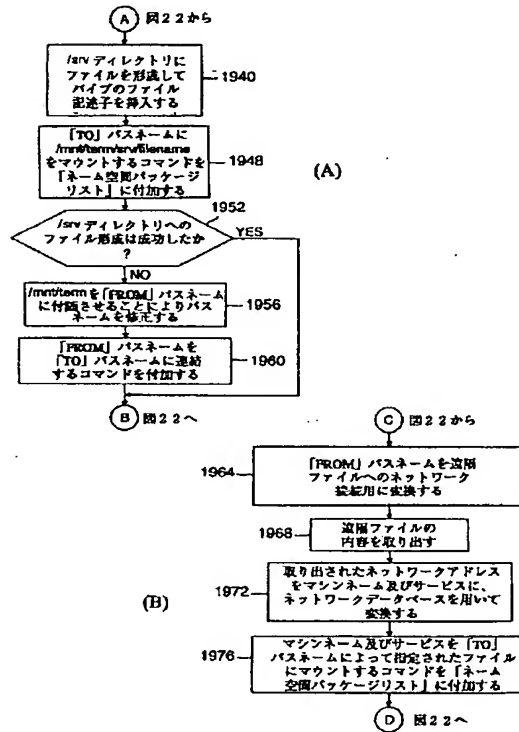
【図21】



【圖 22】



【图 23】



【圖 24】

